

# **Hajautettu ohjelmistotyö, kaksi tapaustutkimusta viestinnästä**

Jussi-Pekka Koro

Tampereen yliopisto  
Informaatiotieteiden yksikkö  
Tietojenkäsittelyoppi / Tietojärjestelmät  
Pro gradu -tutkielma  
Ohjaaja: Mikko Ruohonen  
Joulukuu 2015

Tampereen yliopisto  
Informaatiotieteiden yksikkö  
Tietojenkäsittelyoppi / Tietojärjestelmät  
Jussi-Pekka Koro:  
Pro gradu -tutkielma, 61 sivua  
Joulukuu 2015

---

## **Tiivistelmä**

Tämän tutkielman tarkoituksena on tarkastella nykyään yleistyvän hajautetun ohjelmistokehityksen yleisiä haasteita. Yleisimpiä syitä hajautetun ohjelmistokehityksen ulkoistamiseen ovat mm. taloudelliset syyt (halvempi työvoima), laajemman työmarkkina-alueen takia (osaava henkilöstö), houkuttelevat markkina-alueet (asiakas lähempänä) sekä nopeampi tuotantoaika. Tutkielmassa tarkastellaan erityisesti viestinnän haasteita hajautetussa ohjelmistokehityksessä sekä ammattimaisten käytäntöjen merkitystä hajautetussa ohjelmistokehityksessä. Tutkimuksessa tarkastellaan aluksi kirjallisuudessa hajautetussa ohjelmistotyössä havaittuja ongelmia, joista yksi merkittävimmistä on viestintään liittyvät haasteet. Tutkimuksessa käydään läpi kaksi omakohtaisiin kokemuksiin pohjautuvaa tapaustutkimusta. Tutkimuksessa kootaan myös eräänlaisia työkaluja, joiden avulla mm. viestintää voidaan tehostaa hajautetussa ohjelmistoprojektissa.

Avainsanat ja -sanonnat: Hajautettu ohjelmistotyö, viestintä.

## Alkusanat

Opintojen alussa en juuri vaivannut päätäni mahdollisella valmistumisella. Aika juoksi nopeammin, kuin opinnot etenivät. Näinhän sitä kuulee sanottavan, että aika kuluu nopeasti, kun on mielekästä tekemistä. Monien vaiheiden ja erinäisten toimien jälkeen alkoi kuitenkin näyttämään siltä, että voisin jonakin päivänä valmistua. Kun ei kuuseen kurkottele, niin ei siihen katajaankaan kapsahda. *Pessimisti ei pety*, ei tälläkään kertaa! Opintojen varrelle mahtui sekä onnistumisia että epäonnistumisia. Yksi asia on kuitenkin varma:

*"Kyllä virheet on tehtävä itse, ei puolestani kukaan muu niitä tee."*

[Martikainen, 2015].

Maailma on täynnä yhteen lauseeseen kiteytyneitä viisauksia. Gradua tehdessäni silmiini osui otsikko, joka kuvaa mielestäni hyvin sitä, ettei viestintä todellakaan ole pelkkiä sanoja:

*"Sanat ovat ihmisten välisen kommunikaation turhaa pintakiiltoa."*

[Kaurismäki, 2011].

Lopuksi haluaisin kiittää teitä, jotka ovat suoraan tai välillisesti edesauttaneet opintojeni valmistumisessa, mutta tasapuolisesti myös passiivisesti valmistumistani vastustaneita. Ilman teitä kaikkia tämä työ ei olisi koskaan valmistunut. On loppuhuudon aika: "Hyvä minä, hyvä me!".

Nokia, 25. marraskuu 2015

Jussi-Pekka Koro

## Sisällys

1. Johdanto .....	1
2. Hajautettu ohjelmistokehitys.....	3
2.1 Mitä on hajautettu ohjelmistokehitys? .....	3
2.2 Hajautetun ohjelmistotyön edut ja haitat .....	3
2.3 Viestintä .....	5
2.4 Kulttuuri ja kielimuuri.....	12
2.5 Transaktiivinen muisti, kollektiivinen älykkyys ja ryhmätietoisuus .....	17
2.6 Tietämyksenhallinta ja tiedon jakaminen .....	19
2.7 Ryhmän suorituskky.....	23
2.8 Virtuaalitiimi .....	24
2.9 Joukkoistaminen .....	26
2.10 Ohjelmiston laatu .....	27
3. Tapaustutkimukset.....	35
3.1 Tutkimusmenetelmä ja tutkimusaineisto.....	35
3.2 Tapaus 1: työn siirtäminen kustannussyistä.....	36
3.3 Tapaus 2: yhtenäiset testauskäytännöt virtuaalitiimin avulla .....	42
3.4 Tapaustutkimusten keskinäinen vertailu.....	47
4. Viestinnän tehostaminen hajautetussa ympäristössä.....	49
5. Päätelmät ja yhteenveto.....	56
Viiteluettelo .....	58

## 1. Johdanto

Tämän tutkielman tarkoituksena on tutustua *hajautetun ohjelmistosuunnittelun* (distributed software development) haasteisiin erityisesti viestinnän vaikutuksista kokonaisuuteen. Hajautettu ohjelmistosuunnittelu on nostanut suosiotaan viimeisen vuosikymmenen kuluessa. Ammattitaitoisen henkilöstön tarve, työn kustannusten alentaminen sekä mahdollisuus päästä kilpailluille markkinoille ovat muutamia syitä, miksi ulkoistaminen ja töiden hajauttaminen ovat alati kiinnostavampi aihe kiristyneen kilpailun markkinoilla.

Tämä tutkielma tutustuu kirjallisuuteen liittyen hajautetun ohjelmistotyön haasteisiin ja mahdollisiin ratkaisuihin. Kirjallisuudesta poimittujen tutkimusten tuloksia peilataan kahden omakohtaisiin kokemuksiin pohjautuvien tapaustutkimusten suhteen. Tutkielmassa kuvatut tapaustutkimukset kuvaavat kahta lähtökohdiltaan erityyppistä tapausta.

Ensimmäinen tapaustutkimus voidaan liittää tavoitteeseen vähentää työvoimakustannuksia. Tapaustutkimus ei sinällään ota kantaa siihen, saavutettiinkö töiden siirtämisellä haluttua kustannusten säästöä. Tarkoituksena on kuvata koko työnsiirtoprosessista ainoastaan tietämyksen siirtäminen, joka osaltaan tapahtui kolmen viikon mittaisen ajanjakson aikana. Tämän ajanjakson aikana keskityttiin kasvokkain tapahtuvaan tietämyksen siirtämiseen.

Toisessa tapaustutkimuksessa kuvataan yritys yhtenäistää yrityksen testauskäytäntöjä eri tuotteiden välillä. Tähän tapaustutkimukseen liittyy vahvasti *virtuaalitiimin* (virtual team) käsite.

Viestinnän ongelmia hajautetuissa projekteissa on tutkittu ja miten erilaisia esteitä voitaisiin madaltaa viestinnän näkökulmasta. Tämän tutkielman tarkoitus on koostaa ensin kirjallisuudessa tehdyt havainnot hajautetun ohjelmistosuunnittelun ongelmista viestinnän suhteen. Viestintä-käsite on todella laaja, koska lähestulkoon jokainen vaihe hajautetussa projektissa vaatii viestintää.

Tutkielma ei ota suoraan kantaa projektissa käytettäviin prosesseihin tai menetelmiin vaan pyrkii esittämään vaihtoehtoja viestinnän tehostamiseen, tietämyksen sekä osaamisen tehokkaaseen jakamiseen *maantieteellisesti erillään* (geographic distance) olevien ryhmien välillä.

Tutkimusmetodina on käytetty osittain kirjallisuustutkimusta sekä kahta tapaustutkimuksia.

*Tutkimuskysymys: Mitä ovat hajautetun ohjelmistosuunnittelun ongelmat viestinnässä ja miten niitä voidaan ehkäistä?*

Tutkielma on jaettu seuraaviin osiin: luku kaksi esittää kirjallisuudesta poimittuja hajautettuun ohjelmistokehitykseen liittyviä tutkimuslöydöksiä sekä joukon käsitteitä, jotka liittyvät läheisesti viestintään sekä menetelmiin jakaa osaamista ja tietämystä tehokkaammin koko projektin kesken. Luvussa kolme kuvataan tässä tutkimuksessa käytetty tutkimusmetodi sekä kaksi tapaustutkimuksesta, jotka perustuvat pääosin omiin kokemuksiin työskennellessäni kyseisissä projekteissa sekä matkan varrella saatuihin kokemuksiin. Luvussa neljä pohditaan erilaisia menetelmiä ja käytäntöjä, kutsuttakoon niitä ”työkaluiksi”, taistelussa selättää hajautetun ohjelmistosuunnittelun haasteet erityisesti viestinnän suhteen, sekä luku viisi lyhyen yhteenvedon.

## 2. Hajautettu ohjelmistokehitys.

Tässä luvussa kuvataan mitä hajautetulla ohjelmistokehityksellä oikeastaan tarkoitetaan sekä mitkä ovat sille ominaisia piirteitä. Lisäksi kuvataan valikoituun kirjallisuuteen pohjautuen erilaisia käsitteitä, jotka ovat keskeisiä hajautetulle ohjelmistokehitykselle mm. viestinnän näkökulmasta. Käsitteet ovat yleisesti päteviä myös *keskitetyssä ohjelmistokehityksessä* (centralized software development), mutta hajautetussa ympäristössä niiden vaikutus korostuu enemmän.

### 2.1 Mitä on hajautettu ohjelmistokehitys?

Hajautetulla ohjelmistokehityksellä viitataan yleensä maantieteellisesti hajautettuun ohjelmistokehitykseen, jolloin samaa ohjelmistoa suunnittelee, kehittää ja testaa vähintään kaksi henkilöä ja jotka työskentelevät vähintään kahdessa eri toimipisteessä. Henkilöt voivat edustaa samaa tai eri organisaatiota. Hajautettua ohjelmistokehitystä voidaan pitää tapana tehdä töitä, johon esimerkiksi ohjelmistokehityksen *ulkoistaminen* (outsourcing) johtaa. Ohjelmistokehityksen ulkoistaminen ei ole mikään uusi ilmiö [Lee *et al.*, 2000], mutta se alkoi yleistyä todenteolla vasta 90-luvulla. Ulkoistamisen eri malleja on kuvannut mm. Oshri [2009]. Eri mallit voidaan jakaa sen perusteella, ulkoistetaanko työ tytäryhtiölle vai täysin ulkopuoliselle toimijalle, tehdäänkö työ asiakkaan omissa tiloissa vai ulkopuolisen toimijan tiloissa mikä voi sijaita maantieteellisesti samassa maassa (onshore), lähialueella (naapurimaissa) (nearshore) tai etäämmällä (offshore). Myös ulkoistamisen tasot vaihtelevat tapauskohtaisesti. Ulkoistaminen voidaan toteuttaa esimerkiksi seuraavasti: työ ostetaan kokonaan, työ ostetaan osittain (osa tehdään itse) sekä perustetaan yhteisyritys tai tytäryhtiö. Ulkoistamisen mallin ja tason valintaan vaikuttaa se minkä tyyppistä työtä ollaan ulkoistamassa sekä miten työt aiotaan ulkoistaa.

### 2.2 Hajautetun ohjelmistotyön edut ja haitat

Projektien hajauttaminen on enenevässä määrin alkanut kiinnostamaan yrityksiä. Mitkä ovat syyt siihen, että yritykset ovat kiinnostuneet työn hajauttamisesta tai ulkoistamisesta ja mitkä ovat niiden mahdollisia hyötyjä ja haittoja? Eräässä tutkimuksessa todettiin, että aikaeroista saatava teoreettinen 24 tunnin *kehityssykli* (development cycle) ei ollut syynä ulkoistamisella, vaan sopivan työvoiman parempi saatavuus [Komi-Sirviö ja Tihinen, 2005]. Ulkoistamiseen johtavia syitä etuineen ja haittoineen on tutkittu melko laajasti. Oshri [2009] on todennut, että tietoliikenneyhteyksien parantuminen maailmanlaajuisesti

on pienentänyt etäisyyksien ja ajan vaikutusta tehdä yhteistyötä maailmanlaajuisesti. Toiseksi kehittyvissä maissa on tapahtunut kehitystä, mm. infrastruktuuri on parantunut merkittävästi esim. televiestinnän osalta. Mahdollisia muita syitä jotka saattavat houkutella ulkomaisia yrityksiä ovat muuan muassa mahdolliset verohyödyt, poliittisen ilmapiirin muuttuminen myönteisemmäksi ulkomaisia yrityksiä kohtaan sekä halvemmat työvoimakustannukset. Ainakin osa näistä tarjoaa länsimaisille yrityksille suoria kustannussäästöjä. Myös koulutustason kehittyminen kehittyvissä maissa on luonut vankemman pohjan koulutetulle työvoimalle. Myös tämä on seikka, joka houkuttelee yhä useammin länsimaisia yrityksiä laajentamaan toimintaansa kehittyviin maihin. Talouden kasvu Aasian kehittyvissä maissa kuten Kiinassa ja Intiassa, joiden väkiluku on yhteensä reilut kaksi ja puoli miljardia, tarjoavat valtaiset markkinat myös länsimaisille yrityksille. Ulkomaisten yritysten on tosin verrattain vaikeata päästä näille markkinoille, mutta kynnystä voidaan madaltaa mm. alihankinnan kautta tai perustamalla yhteisyrityksiä lähemmäksi markkinoita. Narayan [2007] on maininnut lisäksi mahdollisuuden työpäivän pidentämiseen aikaeroista johtuvista syistä. Työtä voidaan ainakin teoriassa jatkaa toisella aikavyöhykkeellä, kun toisessa paikassa työpäivä päättyy. Komi-Sirviö [2005] on viitannut aikavyöhykkeistä saatavan edun tarjoavat mahdollisuuden saada tuotteet nopeammin markkinoille sekä mahdollisuuden vastata paremmin paikallisen asiakkaan tarpeisiin.

Kaikella on kuitenkin hintansa, niin myös ulkoistamisella ja hajautetulla ohjelmistokehityksellä. Lukuisten hyötyjen ohella on havaittu olevan myös vähintään yhtä lukuisa määrä ongelmia. Useat tutkimukset ovat havainneet muun muassa seuraavia ongelmia hajautetuissa ohjelmistoprojekteissa. Jiménez ja muut [2009], Bikram ja muut [2006] ja Oshri [2009] ovat luetelleet seuraavia ongelmakohtia: maantieteellinen etäisyys, kulttuurilliset erot, aikaero, kielimuurit, poliittiset, lainsäädännölliset asiat, tietoturva-asiat, työvoimaan liittyvät, sosiaaliset sekä logistiset ongelmat.

Seuraavissa kohdissa käsitellään hieman tarkemmin asioita, joilla keskeinen rooli hajautetun ohjelmistotyön onnistumisen kannalta kuten viestintä, kulttuuri, *tietämyksenhallinta* (knowledge management), *transaktiivinen muisti* (transactive memory). Ryhmän suorituskyky on asia, jolla edellä mainittuja asioita voidaan mitata, asia joka insinööritieteissä on usein kovinkin tärkeää. Virtuaalitiimin sekä joukkoistamisen käsitteisiin tutustutaan hieman, koska toinen tapaustutkimus liittyy virtuaalitiimin käytäntöihin ja haasteisiin. Ohjelmiston laatu kohdassa käydään hyvin yleisellä tasolla mistä ohjelmiston laatu koostuu. Tämä siksi, että ohjelmiston laatu on juurikin se asia, mikä näkyy asiakkaalle, eivät hienoimmatkaan prosessit tai käytännöt joita projektissa käytetään. Myös viestintä voidaan linkittää ohjelmiston laatuun sitä kautta, että erityisesti hajautetuissa ohjelmistoprojekteissa ohjelmiston laadulliset asiat nousevat esiin



testauksesta saatujen tulosten kautta ja miten ne tehokkaasti jaetaan projektin eri kohderyhmille. Myös järjestelmän eri osa-alueiden integrointi ja testaaminen on ominaispiirre, johon hajautetussa ohjelmistotyössä ajaututaan. Myös integrointi vaatii tehokasta viestintää.

## 2.3 Viestintä

Viestintä on keskeinen tekijä ohjelmistoprojektien onnistumisessa. Tässä kohdassa kuvataan ensin mistä viestintä muodostuu organisaation sisällä sekä yleisesti millaista viestintää tarvitaan ohjelmistokehityksessä. Tämän jälkeen kuvataan millaisia erityispiirteitä hajautettu ohjelmistokehitys tuo tullessaan.

Paasivaara [2005] on koonnut kirjallisuudesta organisaation viestinnän kymmenen keskeisintä elementtiä. Alla on listattuna keskeisimmät elementit alkaen tärkeimmästä:

1. Tärkein elementti viestinnässä on itse *viesti* (message), joka sisältää viestitettävän informaation.
2. Viestintään tarvitaan aina vähintään kaksi henkilöä, viestin *lähettäjä* (sender) sekä
3. viestin *vastaanottaja* (receiver).
4. Viestintään on aina jokin *syy* (reason), miksi halutaan viestiä tai on tarvetta välittää viestiä.
5. Viestivien henkilöiden välillä vallitsee *suhde* (relationship).
6. Viestijät voivat kuulua johonkin *viestintäverkostoon* (communication network) esimerkiksi organisaatioon, jonka hierarkkinen rakenne määrittelee mihin viestijä kuuluu ja se taas määrittelee viestinnän suunnan.
7. Viestin *suunnat* (suunnat) voivat olla joko hierarkiassa ylöspäin, alaspäin tai horisontaalia (vertaisviestintä).
8. Viestin välittämiseen tarvitaan jokin *kanava* (media).
9. Viestintään vaikuttaa myös *työympäristö* (environment) sekä
10. viestintään voi vaikuttaa jokin *häiriötekijä* (noise).

Nämä elementit ovat periytyneet osaksi hajautetussa ohjelmistokehityksessä käytävää viestintää.

Ohjelmistokehitys on luonteeltaan sen tyyppistä, että viestinnältä ei käytännössä voida välttyä minkään kokoluokan projekteissa. Kirjallisuuskatsauksessa Jiménez ja muut [2009] ovat tutkineet hajautetun ohjelmistotyön haasteita ja viestintä on eräs niistä. Kirjallisuus katsauksessa todetaan, että koko ohjelmiston kehityskaaren aikana ryhmän jäsenten

viestinnän tarve on suuri. Ryhmän jäsenet käyttävät viestinnässä erilaisia työkaluja (esim. sähköposti) sekä muodoltaan erilaisia viestejä. Tämä aiheuttaa helposti väärinkäsityksiä sekä vasteaikojen pitenemistä. Edelliset ongelmat liitettynä monimutkaiseen infrastruktuuriin sekä alati vaihtuvaan henkilöstöön heikentävät viestinnän tasoa sekä määrällisesti että laadullisesti. Viestinnän määrän väheneminen ja heikko laatu taas saattavat johtaa tuottavuuden heikkenemiseen. Voidaankin yleisesti todeta, että vaikka projektissa työskentelisi yksi henkilö tuotannon alusta loppuun saakka, niin viimeistään asiakkaiden kanssa on jossain vaiheessa viestittävä jonkin median välityksellä. Projektin koon kasvaessa, käytännössä myös projektin henkilöstön määrä kasvaa ellei sitten projektin kestoa haluta kasvattaa ja harvemmin halutaan, toki budjetin sallimissa rajoissa. Henkilöstön määrän kasvaessa myös viestinnän tarve kasvaa.

Brooks [1975] julistaa teoksessaan seuraavaa: "Adding manpower to a late software project makes it later". Ajatus Brooks'n lausahdukset takana viittaa siihen, että maksimi työvoiman määrä riippuu rinnakkain tehtävien itsenäisten tehtävien määrästä. Näin ollen työvoiman lisääminen tietyn määrän jälkeen ei vaikuta positiivisesti projektin etenemiseen, päinvastoin. Henkilöiden lisääminen projektissa lisää myös tarvetta viestintään, koska uudet henkilöt pitää ensin kouluttaa ja perehdyttää tehtäviinsä, jolloin muiden työpanoksesta ainakin osa menetetään siksi aikaa. Vaikka Brooks'n kirja onkin jo päässyt miehen ikään, niin on siinä oleva sanoma edelleenkin ainakin osittain paikkansa pitävä. Brooks'n laki on myöhemmin kokenut kritiikkiä.

Ohjelmistotuotantomallista riippumatta eri kehitysvaiheista kuten suunnittelusta, toteutuksesta ja testauksesta ovat yleensä vastuussa eri henkilöt ja eri vaiheiden välillä on myös riippuvuuksia keskenään. Esimerkiksi suunnitteluvaiheen ja toteutusvaiheen välille muodostuu riippuvuus. Olivatpa suunnitelmat sitten erittäin tarkkoja yksityiskohtaisia dokumentaatioita tai sitten ainoastaan muutamalla ranskalaisella viivalla kirjattuja vaatimuksia, edeltää se poikkeuksetta toteutusvaihetta. Tosin kovin usein vaatimukset vaativat päivittämistä, jos toteutusvaiheessa havaitaan ongelmia tai puutteita. Ohjelmistokehitys on luonteeltaan dynaamista, joka tarkoittaa sitä, että tarkastikaan tehdyt suunnitelmat eivät yleensä kestä muuttumattomina projektin alusta loppuun saakka [Cataldo and Herbsleb, 2008]. Tästä taas seuraa se, että vaatimusten ja/tai määritysten muuttuessa on todennäköisesti tarvetta tarkistaa projektin aikataulullisia asioita sekä tiedottaa asianosaisia (suunnittelijat, testaajat jne.) muutosten johdosta. Muutoksilla saattaa olla myös muutoksesta riippuen riippuvuus asiakkaan suuntaan, jolloin muutoksista on viestittävä myös asiakkaalle. Projektin koko henkilöstön työskennellessä fyysisesti lähekkäin, esimerkiksi samalla toimistolla, on viestintä muiden kanssa yleensä mutkatonta.

Komi-Sirviö ja Tihinen [2005] ovat todenneet kasvokkain käytävien keskusteluiden ja neuvotteluiden vähenevän selkeästi, kun osa työskentelee eri kaupungissa tai useassa eri kaupungissa. Kasvokkain käytävät tapaamiset vähenevät lähes olemattomiin, kun työskennellään eri maissa ja/tai eri maanosissa. Etäisyys vähentää halua ja mahdollisuuksia järjestää tapaamisia [Komi-Sirviö ja Tihinen, 2005]. Tosin viestintään tai sen vähyyteen vaikuttavat myös muutkin asiat kuin pelkkä fyysinen etäisyys.

Espinosa ja muut [2007] ovat todenneet, että haasteet tulla tutuiksi muiden kanssa sekä tulla tietoisiksi heidän työskentelytavoistaan ja osaamisestaan, jos tätä tietoa ei ole, on vaikeampaa olla yhteydessä toisiin. Esimerkki tämän kaltaisesta tilanteesta voi olla vaikkapa projekti, jossa henkilöstö ei ole toisilleen ennestään tuttuja. Tämä voi johtua esimerkiksi siitä, että osa henkilöistä tulee oman organisaation sisältä ja osa henkilöistä tulee alihankintayrityksestä. Tässä tapauksessa viestinnän esteenä tai hidasteena saattaa olla se, että projektin henkilöstö ei ole ennestään toisille tuttuja, *yhteisen maaperän* (common ground) puuttuminen sekä erilaiset organisaatiokulttuurit. Komi-Sirviön ja Tihisen [2005] mukaan pohja keskinäiselle luottamukselle ja yhteenkuuluvuuden tunne luodaan projektin alussa. Yhteenkuuluvuuden tunne tulisi olla sillä tasolla, ettei mitään esteitä ole ottaa toiseen yhteyttä. Tämän takia projektin alussa kannattaisi kiinnittää huomiota siihen miten projektiryhmä toimii keskenään ja puuttua ongelmiin mahdollisimman nopeasti.

Kuten yleensä, ongelmien korjaaminen aikaisessa vaiheessa, säästää aikaa ja rahaa sekä varmistaa ettei ongelmista pääse muodostumaan todellisia ongelmia projektin edetessä. Pienistäkin ongelmista saadaan kehitettyä suuren luokan kriisejä, jos niihin ei osata puuttua ajoissa. Aiempien tutkimusten mukaan avokonttorin uskotaan yleisesti helpottavan viestintää sekä yhteistyötä työkavereiden kesken, jonka pitäisi nostaa yksilön tehokkuutta sekä yrityksen tuottavuutta [Kim and de Dear, 2013]. Tästä asiasta on kuitenkin kovin vähän kokemusperäisiä todisteita [Kim and de Dear, 2013]. Kun asiaa mietitään pelkästään viestinnän kannalta, niin avokonttori on mitä mainioin tapa vapauttaa spontaani viestintä ryhmän jäsenten kesken. Tämä on erityisen hedelmällistä ongelmatilanteiden vaatiessa tiivistä yhteistyötä, esimerkiksi vaatimusmäärittelyä tekevän henkilön ja toteutusta tekevän henkilön välillä tai toteuttajan ja testaajan ratkaistessa ongelmia. Samalla jokainen on tiedostamattaankin tietoinen siitä ketä tilassa on läsnä (läsnäolotietoisuus) [Espinosa *et al.*, 2007]. Jokainen henkilö myös väistämättä rekisteröi muiden tekemiset ja keskustelut, vaikka ei sitä itse tiedostaisi. Tämän etuna voidaan pitää sitä, että melkeinpä huomaamattaan jokaisella on melko hyvä käsitys projektin tilasta (tehtävätietoisuus) [Espinosa *et al.*, 2007]. Koko työryhmän läsnäolo helpottaa myös palaverien järjestämistä spontaanisti, kunhan vaan työtilat on suunniteltu oikein. Varjopuolena voidaan taas pitää muun muassa avokonttorissa vallitsevaa jatkuvaa

meteliä ja keskittymistä vaativan työn vaikeutumista sekä työtehokkuuden laskemista tai ainakin tunnetta siitä, ettei saa mitään aikaan edellä mainittujen asioiden takia [Kim and de Dear 2013].

Hajautetussa ohjelmistokehityksessä kaikki haasteet ovat periaatteessa samoja kuin keskitetyssä ohjelmistokehityksessä, mutta seuraavat asiat aiheuttavat osaltaan erityisiä haasteita viestinnän suhteen, kuten fyysinen etäisyys, aikaero, kulttuurierot tai kielimuuri.

Carmel [2005] on listannut viisi etäisyyden aiheuttamaa ongelmaa hajautetussa ohjelmistotyössä: Viestinnän puute, koordinoitioongelmat, kontrollin puute, eheyden puute ja. kulttuurien törmäys.

Fyysinen etäisyys (maantieteellinen hajautuneisuus) rajoittaa viestintävälineet käytännössä sähköisiin viestimiin kuten *sähköposti*, *pikaviestimet*, *keskustelupalstat*, *puhelin*, *konferenssipuhelut* ja *videopuhelut*. Alla on kuvattu lyhyesti kuhunkin viestimeen liittyvistä hyvistä ja huonosta puolista.

Sähköpostia käyttäessä viestin sisältöä voidaan tulkita helposti väärin. Väärin tulkintaan saattavat johtaa esimerkiksi seuraavat tekijät: Henkilöiden käyttämän termistön erilaisuus, vieraalla kielellä kirjoitettu viesti saattaa sisältää asiavirheitä tai kirjoittaja saattaa olettaa kysymyksessään, että vastaanottaja tuntee sovellusalueen yhtä hyvin kuin hän, jolloin viestistä saatetaan jättää oleellisiakin asioita pois. Tämä saattaa johtaa siihen, että vastaanottaja ei ymmärrä kysymystä tai hän tulkitsee viestin väärin. Yhtenä sähköpostin heikkoutena voidaan pitää myös sitä, että viestiin ei välttämättä saada nopeata vastausta tai ei saada vastausta ollenkaan. Syitä tähän voi olla, ettei vastaanottaja ei ole ymmärtänyt kirjoittajan kysymystä, hän ei pidä asiaa tärkeänä, asia ei koske suoraan häntä tai kyseinen henkilö ei enää työskentele koko projektissa. Hyvänä puolena sähköpostin käytössä ovat muun muassa se, että viestin voi kirjoittaa rauhassa milloin se kirjoittajalle parhaiten sopii. Sähköpostiviestinnän asynkroninen luonne mahdollistaa viestien lähettämisen vaikka vastaanottaja ei sillä hetkellä olisi tavoitettavissa. Viesteistä jää myös aina jokin ”jälki”, koska viestit yleensä taltioidaan ja näin ollen vanhoihin viesteihin voidaan palata jälkepäin.

Pikaviestimien hyvä puoli on sen nopeus, koska se on luonteeltaan synkroninen. Pikaviestimiin sisältyy usein jonkinlainen indikaattori henkilöiden paikallaolo-statuksesta, jolloin suurella todennäköisyydellä viestiin vastataan nopeasti, jos henkilö on paikalla. Hyvänä asiana voidaan pitää myös sitä, että viestijöiden on mahdollisuus esittää heti tarkentavia kysymyksiä ja näin ollen mahdollisilta väärin ymmärryksiltä voidaan paremmin välttyä. Nykyään useista pikaviestimistä löytyy lisäominaisuutena myös

näytönjako jonka avulla voidaan käydä läpi konkreettisemmin ja tarkemmin vaikkapa koodissa havaittuja ongelmakohtia. Näin välttään myös paremmin väärin ymmärryksiltä sekä se on nopeampi tapa kuin viestien edestakainen lähettely. Pikaviestimien heikkoutena voidaan pitää esimerkiksi liiallisen viestinnän aiheuttamaa viestiähyä. Suoraan toiselle lähetettävien viestikanavien lisäksi perustetaan usein viestiryhmiä, joiden kautta voidaan esittää kysymyksiä laajemmalle kohdeyhteisölle. Viestiryhmän ongelmana saattaa olla se, että ryhmässä vastuu on jakautunut koko ryhmän kesken ja näin ollen hankalat tai huonosti muotoillut kysymykset jäävät vaille huomiota. Ryhmien määrän kasvaessa, viestien määrä luonnollisesti kasvaa ja käyttäjältä saattaa jäädä huomioimatta tärkeätkin viestit. Myös asioiden etsiminen jälkeinpäin viestiketjuista saattaa olla hankalaa tai jopa mahdotonta. Omana alahaarana pikaviestinten joukossa on pitänyt pintansa merkkipohjainen IRC [IRC, 2015].

Keskustelupalstat tarjoavat keskustelualueita, johon saadaan koottua samasta aiheesta kiinnostuneita henkilöitä. Keskustelupalstan käyttötarkoitus on hieman vastaava kuin pikaviestimen viestiryhmillä. Koska keskustelupalstat eivät ole organisaatioiden sisällä useinkaan ensisijaisia viestintäkanavia, voidaan olettaa, että juuri tämän vuoksi niiden kautta olisi mahdollista tavoittaa aiheeseen vihkiytyneitä henkilöitä. Keskustelupalstojen tarkoituksena on tavoittaa mahdollisimman laaja kohdeyleisö mahdollisimman helposti. Keskustelupalstojen viestiketjut säilyvät myös paremmin kuin pikaviestinten ja näin ollen vanhojen asioiden etsiminen saattaa olla hieman helpompaa, joskaan ei välttämättä helppoa. Keskustelupalstat tarjoavat esimerkiksi oivan kasvualustan testauskäytäntöjen levittämiseen organisaation sisällä.

Puhelimen käytön etuna voidaan pitää sen reaaliaikaista luonnetta. Suurella todennäköisyydellä molempien osapuolien huomio on keskittynyt ainoastaan käytävään keskusteluun. Puhelimessa käytyjen keskustelujen heikkouksina voidaan pitää mahdollisesti seuraavia asioita: heikko puhelinyhteys esimerkiksi konferenssipuheluissa, heikot viestit, jotka eivät välity pelkässä puheessa, puhelut harvemmin tallentuvat, jolloin keskustelusta ei jää automaattisesti jälkeä, kynnys soittaa tuntemattomalle henkilölle, tuntemattomille soittajille ei välttämättä vastata sekä pitkästä aikaerosta johtuen yhteisen ajan löytäminen saattaa olla hankalaa.

Konferenssipuhelut ovat hyvä tapa saada useampi henkilö samaan aikaan koolle. Suuremmalla foorumilla on todennäköisempää, että asiantuntijuutta eri asioihin löytyy enemmän ja sitä myöden vastauksia ongelmiin löytyy helpommin. Konferenssipuheluiden yhteyteen yhdistetään myös yleensä tietokoneella jaettu ja seurattava viestintää tukeva esitys. Heikkouksia ovat kuten myös puhelimen käytössä heikko äänenlaatu sekä alati vaivaavat tekniset ongelmat kuten vaikeuksia saada yhteyttä, ongelmia näytönjaossa jne.

Myös osallistujien määrällä on vaikutusta keskusteluun, mitä enemmän osallistujia, sitä vaikeampaa on seurata keskustelua ja tietää kuka milloinkin on äänessä. Keskittymistä saattaa haitata myös se, että käytävä keskustelu tai aihe ei aina koske kaikkia osallistujia ja siitä saattaa seurata osallistujien seuraamisen herpaantuminen.

Videopuheluiden etuna on se, että henkilöt näkevät kenen kanssa keskustelevat ja eleet välittyvät mahdollisesti paremmin kuin pelkän äänen välityksellä. Heikkouksina voidaan pitää ainakin mahdollisia ongelmia tietoliikenneyhteyksissä, josta johtuen kuva ja ääni eivät välity parhaalla mahdollisella tavalla. Tekniset ongelmat laitteissa aiheuttavat myös ongelmia. Heikkoutena voidaan pitää myös sitä, että korkeatasoisiin videopuheluihin tarvittavaa laitteistoa ei ole aina saatavilla.

Kaikille näille sähköisille viestimille yhteinen piirre on se, että ne eivät sisällä kaikkia viestejä mitä kasvotusten käytävä keskustelu pitää sisällään. Erran ja Tjia [2005, s.151] on todennut, että jopa 80 prosenttia viestinnästämme perustuu muuhun kuin kirjoitettuun tekstiin. Näin ollen ainoastaan videopuhelut sisältävät lähestulkoon kaikki viestinnässä käytetyt elementit. Puhelut taas eivät sisällä kehonkielen ilmaisemia viestejä. Sähköpostista ja muissa tekstipohjaisista viestimenetelmissä on käytössä ainoastaan teksti.

Yksi suurimmista ongelmista ja aikaa vievin tekijä virheitä korjattaessa oli aikavyöhykkeiden aiheuttamat aikaerot [Erran and Tjia, 2005, s.160]. Aikaero aiheuttaa viestinnän kannalta ongelmia muun muassa yhteisen työajan vähyyden vuoksi. Tämän vuoksi esimerkiksi yhteisten palaverien järjestäminen molemmille sopivaan ajankohtaan tuottaa usein ongelmia. Palaverien määrä on kuitenkin todennäköisesti jopa suurempi hajautetuissa projekteissa, koska käytännössä viestinnän tarve on suurempi. Näin ollen, jos palaverien määrää ei voida pienentää, niin silloin palaverit on järjestettävä toisen osapuolen kannalta sopimattomiin aikoihin. Oletetaan, että projekti on hajautettu kolmelle eri mantereelle Aasiaan, Eurooppaan sekä Pohjois-Amerikkaan. Aikavyöhykkeet ovat seuraavat: Suomi (Eurooppa) +2, Yhdysvallat (Pohjois-Amerikka) -10(-5) sekä Kiina (Aasia) +6(+8), kesäaikaa ei huomioida. Pahimmillaan aikaero voi olla Suomen näkökulmasta 8 tuntia toisiin maihin. Jos palaveri järjestetään esimerkiksi kello 14 Suomen aikaan, niin palaveri Yhdysvalloissa alkaa kello 6 ja Kiinassa kello 22. Aikaeroa on tosin myös markkinoitu hajautettujen projektien kannalta edullisena asiana, koska ainakin periaatteessa työtä voidaan jatkaa tauotta kellon ympäri. Aikaerolla on myös merkitystä, jos ajatellaan ongelmanratkaisu tilanteita. Jos viestintään käytettävä media on esimerkiksi pelkästään kirjoitettuun sisältöön pohjautuva, esimerkiksi sähköposti, niin monen asian pitää onnistua, että ongelma saadaan ratkaistua sujuvasti. Varmasti yleisin syy ongelmiin sähköpostin välityksellä käytävässä viestinnässä on puutteelliset tai epäselvät viestit. Mahdollisia syitä viestinnän epäselvyyksiin ovat: viestin sisällön virheelliset tulkinnat,

viestin sisältö on puutteellinen, viesti lähetetään henkilölle jolla ei ole pätevyyttä vastata, viesti lähetetään liian suurella jakelulla ja viestiin ei vastaa kukaan.

Aikaeron hallitsemiseen muun muassa Erran ja Tjia [2005, s. 160] on ehdottanut seuraavia tapoja lieventää aikaeroon liittyviä ongelmia. Aikaerosta riippuen ryhmien yhteinen työaika vaihtelee melkoisesti. Pahimmillaan ryhmillä ei käytännössä ole yhteistä työskentelyaikaa juuri lainkaan, jos molemmat työskentelevät normaaleihin toimistoaikoihin. Myös tietoisuus toisen ryhmän normaaleista työajoista ja kansallisista vapaista yms. ennaltaehkäisee mahdollisilta yllätyksiltä. Seuraavassa on listattu Erran ja Tjia [2005] ehdottamia parannuksia niihin ajankohtiin, jolloin ryhmät eivät ole yhtä aikaa töissä, yhteisellä työajalla sekä mitä on syytä tietää toisen ryhmän työajoista.

Aikana jolloin toinen ryhmä ei ole reaaliaikaisesti tavoitettavissa, voi ryhmä tämän ajan keskittyä normaalisti omien työtehtäviensä tekemiseen. Halutessaan välttää reaaliaikaista viestintää tulee yhteistyölle olla selkeät raamit, tehtävien tulee olla selkeästi määriteltyjä, tehtäville tulee määritellä vastuullinen henkilö sekä toimitettavat kokonaisuudet on myös määriteltävä riittävän tarkasti. Suunnittelemalla päivän työt jo etukäteen voidaan yhteinen työskentelyaika käyttää tehokkaasti pakollisten palaverien järjestämiseen. Parhaimmillaan, kun tehtävät on suunniteltu hyvin, esimerkiksi yhden päivän kokoisiksi, voidaan omat tuotokset toimittaa ennen työpäivän päättymistä toisen ryhmän käyttöön. Ainakin teoriassa, jos kaikki menee suunnitellusti, tarve viestinnälle vähenee ja työt tulee tehtyä ajallaan.

Erilaisia tapoja kasvattaa yhteisen työajan määrää on muutamia. Yleisin tapa on joustaa työajoissa. Käytännössä se tarkoittaa sitä, että esimerkiksi Euroopassa työskentelevät henkilöt siirtävät työpäivän alkua myöhemmäksi sekä Amerikassa työskentelevät henkilöt vastaavasti aloittavat työpäivänsä normaalia aikaisemmin. Tätä tapaa voidaan käyttää jopa säännöllisesti, jos järjestelyt sopivat työntekijälle tai vastaavasti ainoastaan silloin, kun on tarvetta tehostetulle reaaliaikaiselle viestinnälle esimerkiksi projektin kriittisessä vaiheessa. Toinen tapa on joustaa vielä enemmän työajoissa ja tehdä reilusti pidempää työpäivää tai olla tavoitettavissa käytännössä koko valveillaoloajan. Työssäjaksamisen kannalta tämä vaihtoehto ei ole optimaalisin, mutta tarvittaessa väliaikaisena ratkaisuna tavoitteiden saavuttamiseksi se saattaa olla tarpeen. Kolmantena vaihtoehtona on mahdollisuuksien mukaan vuorottaa työntekoa esimerkiksi siten, että osa työntekijöistä aloittaa päivänsä normaalia aikaisemmin ja osa taas vastaavasti normaalia myöhemmin. Täten ainakin osalla työntekijöistä on mahdollisuus reaaliaikaiseen viestintään näin tarvittaessa. Neljäntenä vaihtoehtona on se, että vain henkilöt joiden työtehtävien takia on joka tapauksessa tarve olla reaaliaikaisesti yhteydessä etäälle, ottavat viestinvälittäjän roolin ja pitävät ryhmät ajan tasalla toistensa tekemisistä. Tähän rooliin yleensä omasta

vapaasta tahdostaan tai roolinsa perusteella joutuvat projektipäälliköt, ryhmäpäälliköt tai henkilöt joilla on muita suurempi vastuu viestinnästä. Hyväksi todettu tapa ongelmatilanteessa on havaita riittävän ajoissa, jos ei-reaaliaikainen viestintä ei johda haluttuun lopputulokseen, ottaa puhelin käteen ja soittaa.

Viimeisenä on hyvä sisäistää aikaeron vaikutus käytännön asioihin. Palaverin järjestäminen itselle sopivaan aikaan voi tarkoittaa palaverin alkamista toiselle epäsuotuisaan aikaan. Samoin on hyvä muistaa mihin aikaan on soveliaista soittaa. Kokemuksen myötä asiasta tulee itsestään selvyys, mutta hajautetussa työssä ensi metrejä ottavat saattavat asian innostuksissaan unohtaa.

Työajoissa on kansallisia eroja [Erran and Tjia, 2005]. Eri puolilla maailmaa on erilaisia käytäntöjä normaalista työajasta. Samoin lounastaukoja pidetään eri aikoihin tai niitä voidaan pitää kahdesti, kuten Kiinassa on tapana. Eurooppalaisille on itsestään selvyys, että viikonloppu on lauantai sekä sunnuntai. Näin ei kuitenkaan ole kaikkialla maailmassa, vaan esimerkiksi Israelissa viikonloppu on perjantai ja lauantai sekä arabimaissa perjantai sekä lisäksi joko torstai tai lauantai.

## 2.4 Kulttuuri ja kielimuuri

Kulttuurieroista puhuttaessa tarkoitetaan yleensä eri kansallisuuksien välisiä eroja, mutta myös eri organisaatioiden välisiä kulttuurieroja sekä ammattien välisiä kulttuurieroja. Ehkäpä tunnetuin kulttuurien välisiin eroihin liittyvä tutkimus on psykologi Geert Hofsteden tekemä tutkimus kansallisten kulttuurien ulottuvuuksista. Hofsteden tutkimukset nostavat esille kaikkiaan kuusi eri ulottuvuutta. Lehto [2012] on tiivistänyt hyvin Hofsteden kulttuuriulottuvuudet omassa tutkielmassaan. Hofstede määritteli ensin neljä kulttuuriulottuvuutta ja lisäsi myöhemmin viidennen ja kuudennen. Hofsteden kulttuuriulottuvuudet ovat: 1. *Valtaetäisyys* (power distance). 2. *Individualismi vs. Kollektivismi* (individualism vs. collectivism). 3. *Maskuliinisuus vs. feminiinisyys* (masculinity vs. femininity). 4. *Epävarmuuden välttäminen* (Uncertainty avoidance). 5. *Pitkän vs. lyhyen aikavälin orientaatio* (long-term orientation vs. short-term orientation) ja 6. *Hemmottelu vs. pidättyvyys* (indulgence vs. restraint). Seuraavassa lyhyt kuvaus kustakin Hofsteden kulttuuriulottuvuudesta.

Valtaetäisyysindeksi kuvaa miten organisaatiossa vallan käytön kohde sopeutuu siihen, että valta ei ole tasaisesti jakautunutta ja organisaatiossa on selkeä hierarkia. Pienen valtaetäisyyden maissa, kuten useissa Euroopan maissa organisaatioiden hierarkia ei ole niin selkeä ja se näkyy muun muassa siinä, että lähestulkoon missä tahansa tehtävässä työskentelevä henkilö voi lähestyä organisaation johdossa olevaa henkilöä. Esimerkkinä



voidaan ottaa tilanne, jossa hajautetussa projektissa viestinnän osapuolien tulisi olla hierarkkisesti samalla tasolla, muuten on suuri todennäköisyys siihen, että viestintä loppuu lyhyeen. Valta pienen valtaetäisyyden maissa on hajaantunut organisaatiossa laajemmin sekä palkkaerot eivät ole niin suuria kuin suuren valtaetäisyyden maissa. Asia johon varmasti törmää hajautetuissa projekteissa on oma-aloitteisuuden käsite. Suuren valtaetäisyyden maissa työntekijät odottavat, että hierarkiassa ylemmällä tasolla olevat henkilöt määrittelevät mitä työntekijän tulee milloinkin tehdä, kun taas pienen valtaetäisyyden maissa työntekijä saattaa oma-aloitteisesti tehdä päätöksiä omia töitä koskevista asioista. Ongelmatilanteet ovat esimerkki siitä miten valtaetäisyys vaikuttaa työntekijään. Pienen valtaetäisyyden maissa ongelmia saatetaan alkaa ratkoa oma-aloitteisesti kollegoiden kesken, kun taas suuren valtaetäisyyden maissa oma-aloitteinen uuden määrittelemättömän työtehtävän aloittaminen ei ole soveliasta.

Individualismi vs. kollektivismi-indeksi kuvaa, miten yksittäinen henkilö näkee itsensä osana organisaatiota, yhteisöä tai perhettä. Pienen valtaetäisyyden maat yhdistetään usein individualismiin ja suuren valtaetäisyyden maat kollektivismiin. Yksinkertaistettuna individualistinen ihminen asettaa omat päämääränsä yhteisön päämäärien edelle ja kollektivistinen ihminen taas asettaa yhteisön päämäärät omien päämääriensä edelle.

Maskuliinisuus vs. feminiinisyys indeksi kuvaa sitä, miten yhteisö suhtautuu miesten ja naisten välisiin eroihin ja minkä tyyppisiä arvoja kumpikin sukupuoli yleisesti arvostaa. Naisten katsotaan arvostavan enemmän pehmeitä arvoja kuten ihmissuhteita, ympäristöä jne., kun taas miesten katsotaan arvostavan enemmän kovia arvoja kuten rahaa, valtaa jne.

Epävarmuuden välttämisen indeksi kuvaa sitä, miten yksilö tuntee epävarmuutta ennestään tuntemattomia tilanteita kohtaan ja miten hän yrittää niitä hallita. Epävarmuuden välttäminen liittyy päätöksentekoon ja siihen liitetään sääntöjä joiden taakse on helppo piiloutua. Suuremman epävarmuuden välttämisen maissa oudoista ja epävarmoista tilanteista ahdistutaan enemmän kuin pienen epävarmuuden välttämisen maissa.

Pitkän vs. lyhyen aikavälin orientaatioindeksissä pitkän aikavälin orientaatiolla viitataan sellaisiin asioihin, joiden oletetaan tuottavan hedelmää pitkällä aikavälillä, kuten sitkeys, yrittäminen ja säästäminen. Lyhyen aikavälin orientaatiolla viitataan perinteiden kunnioittamiseen, kasvojen säilyttämiseen sekä sosiaalisten velvoitteiden täyttämiseen.

Hemmottelu vs. pidättyvyys-indeksi kuvaa subjektiivisen hyvinvoinnin eli onnellisuuden määrää. Toisessa ääripäässä onnellisuus saavutetaan korostamalla omaa

vapaudentunnetta, vapaa-ajan merkitystä ja tarvetta elää haluamallaan tavalla. Toisessa ääripäässä taas onnellisuus on kytkettynä tiukkoihin sosiaalisiin normistoihin.

Hofsteden tulkinta on myöhemmin kohdannut vastustusta muun muassa siitä syystä, että siinä kulttuurilla viitataan ainoastaan kansalliseen kulttuuriin. Siinä kulttuuria pidetään myös muuttumattomana ja kulttuurin edustajia homogeenisenä ryhmänä. [Boden *et al.*, 2009].

Erran ja Tjia [2005, s. 176-180] ovat myös listanneet omassa tutkimuksessaan kulttuurisia orientaatioita, osa Hofsteden määrittelemiä ja osa muiden yhteiskuntatieteilijöiden määrittelemiä. Seuraavassa on listaus muista orientaatioista lyhyen kuvauksen kera.

Viestintäorientaatiolla kuvataan miten henkilö viestii ja tulkitsee muiden viestejä. Viestintäkulttuurit voidaan jakaa kahteen eri kategoriaan, korkean kontekstin ja matalan kontekstin kulttuureihin. Matalan kontekstin kulttuurin maissa viestintä perustuu pääosin sanalliseen sisältöön (suora viestintä) ja korkean kontekstin maissa viestin tulkintaan käytetään sanojen lisäksi myös vallitsevaa tilannetta ja fyysisen ympäristön tuomia piirteitä (epäsuora viestintä). Korkean kontekstin kulttuureissa sanat eivät siis kerro kaikkea, vaan kuulijan on osattava tulkita viestistä sekä asiayhteydestä koko asiasisältö. Matalan kontekstin kulttuureissa taas asiat jotka ovat itsestään selvyyksiä korkean kontekstin kulttuureissa ilman sanallista ilmaisua, pitää matalan kontekstin kulttuurissa kuvata tarkasti sanallisesti. Pohjois-Euroopan ja Pohjois-Amerikan katsotaan olevan esimerkkejä matalan kontekstin kulttuurin maista, kun taas esimerkiksi Etelä-Euroopan maita sekä Aasian maita voidaan pitää korkean kontekstin kulttuurin edustajina. Tämän vuoksi matalan kontekstin maissa on luontevampaa käyttää viestintään esimerkiksi sähköpostia. Korkean kulttuurin maissa kirjoitetun sanan rinnalle kaivataan kirjoitetusta sanasta poisjäänyttä tietoa. [Erran and Tjia, 2005].

Kohtalo-orientaatiolla kuvataan, miten henkilö uskoo tapahtumien olevan joko ennalta määrättyjä tai sitten tapahtumat ovat henkilön omien toimien aikaan saamia. Syy-seuraussuhde tulkitaan eri lailla uskooko henkilö tapahtumien ja henkilökohtaisten ominaisuuksien olevan ennalta määrättyjä vai ei. [Erran and Tjia, 2005].

Universalismi-orientaatio kuvaa, miten säännöt ovat universaalisti päteviä ja niiden pitäisi koskea kaikkia tilanteesta riippumatta. Vastakkainen orientaatio on partikularismi, joka ottaa huomioon pelkkien sääntöjen sijaan myös henkilöiden väliset suhteet. Esimerkkinä Erran ja Tjia [2005] esittävät tutkimuksen, jossa kerrottiin alisuorittajasta, joka usean vuoden ajan oli palvellut tunnollisesti työnantajaa. 75 prosenttia amerikkalaisista ja kanadalaisista vastaajista olisivat olleet valmiit päättämään työntekijän työsuhteen, kun

taas vain 20 prosenttia singaporelaisista ja korealaisista olisi ollut valmiit samaan. [Erran and Tjia, 2005].

Melko uusi tutkimus kuvaa, miten erilailla länsimaalaiset ja itä-aasialaiset käsittelevät tietoa. Tutkimuksessaan Nisbett [Erran and Tjia, 2005] on havainnut, että itä-aasialaiset (japanilaiset, kiinalaiset, korealaiset) näkevät yhteyksiä, kun taas länsimaalaiset mieluummin luokittelevat asioita. Esimerkkinä Erran ja Tjia [2005, 180] mainitsevat tilanteen, jossa japanilaisille ja amerikkalaisille näytettiin kuvia akvaariosta. Japanilaiset muistivat 60 prosenttia enemmän taustalla olevia asioita kuin amerikkalaiset. Japanilaiset yhdistivät muistellessaan näkemäänsä kohteeseen jotain taustalla olevaa.

Eräs näkyvä kansallisen kulttuurin elementti on kieli. Hajautetussa ohjelmistokehityksessä viestinnän tarve projektin onnistumiseksi on merkittävässä roolissa. Etäisyyden, aikaeron ja kielimuurin ansiosta viestinnässä on helpompaa käyttää kirjoitetussa muodossa olevaa viestintää kuten sähköposti, pikaviestimet jne. Yhteisenä kielenä on usein käytetty englantia tai jotain muuta yhteisesti tunnettua kieltä. Kieli voi olla kuitenkin molemmille sellainen, jota he eivät käytä äidinkielenään. Tämä aiheuttaa sen, että käytössä olevan sanaston määrä voi olla melko suppeakin. Tämä taas johtaa siihen, että ilmaisuista tulee suppeampia ja väärinkäsitysten mahdollisuus kasvaa. Myös käytettyjen termien erilaiset tulkinnat voivat johtaa väärinymmärryksiin. Tämä saattaa johtua erilaisista kulttuureista, niin organisaation, kansallisen kuin ammatillisenkin. Puhutussa viestinnässä yhteisen sävelen löytäminen saattaa viedä aikaa. Tämä johtuu ainakin osaltaan puhujan omalle äidinkielelle omintakeisen aksentin käyttämisestä sekä sen ymmärtämisen vaikeudesta kuulijalle. Myös puhutussa viestinnässä sanaston suppeus ja vielä molempien osapuolten käyttämä erilainen sanasto aiheuttaa helposti väärinkäsityksiä. Viestijöiden erot osaamistasoissa ja osaamisalueessa voivat myös helposti aiheuttaa väärinymmärryksiä.

Osa edellä mainituista ongelmia voi toki ilmetä myös ei hajautetussa ympäristössä, mutta silloin mahdollisuus viestiä kasvotusten lievittää mahdollisia ongelmia. Näitä asioita ovat muun muassa kielimuuri, kulttuuriset erot ja erot osaamistasossa ja osaamisalueessa.

Melkeinpä intuitiivisesti voisi todeta, että kielellä on merkittävä vaikutus viestinnän onnistumiseen tai epäonnistumiseen. Sekä kulttuuri että kieli ovat vahvasti sidoksissa toisiinsa. On ymmärrettävää, jos viestinnässä käytetään molemmille henkilöille vierasta kieltä, niin silloin käytettävä kieli on todennäköisesti niukka sanaisempaa kuin se olisi käytettäessä omaa äidinkieltä. Tosin esimerkiksi amerikkalaisilla ja englantilaisilla saattaa syntyä väärinkäsityksiä, vaikka kieli on käytännössä sama. Haasteita aiheuttavat kulttuurille ominaiset tulkinnat eri ilmaisuille joita brittiläiset mielellään käyttävät.

Seuraavassa muutamia esimerkkejä kirjasta *Offshoring Information Technology* [Erran and Tjia, 2005]:

”I hear what you say” – tällä amerikkalainen ymmärtää, että britti hyväksyy hänen kantansa asian suhteen. Todellisuudessa britti on asiasta eri mieltä, eikä halua keskustella asiasta enempää.

”Not bad” – tämän amerikkalainen tulkitsee enemmänkin negatiivisena ilmauksena ja ymmärtää asian olevan huono tai keskiverto. Todellisuudessa britin oli tarkoitus hieman kehaista amerikkalaista. Vastaavanlaisia ilmauksia löytyy myös todella paljon lähempää. Kotoisasta suomesta löytyy paikkoja joissa suurin kehu saatetaan ilmaista esimerkiksi seuraavasti: ”Ennemmin tän söi, kun turpaansa otti.”

”Quite good” – Tämän amerikkalainen tulkitsee positiivisena ilmauksena, kun taas brittiläisen tarkoitus oli kiertäen ilmaista, että ei ollut lainkaan tyytyväinen.

”Very interesting” – Amerikkalainen tulkitsee tämän sanonnan positiiviksi palautteeksi ja on sitä mieltä, että britti on aivan otettu ehdotuksesta. Britin tarkoitus oli tällä sanonnalla olla eri mieltä asiasta tai ilmaista epäuskoa asiaa kohtaan.

”I am sure it is my fault” – tällä tavoin brittiläinen ilmaisee kierrellen, että kyllä se todella oli sinun vikasi. Amerikkalainen taas ottaa asian kirjaimellisesti ja pitää sitä heidän vikanaan.

”You must come to dinner sometime” – tällä kohteliaisuudella brittiläinen ei todellisuudessa tarkoita mitään muuta kuin esittää kohteliaisuuden. Amerikkalainen taas tulkitsee asian, niin että hän vastaanottaa kohta kutsun illalliselle.

Tässä vain muutama esimerkki jotka kuvastavat hyvin miten helposti väärinymmärryksiä syntyy, vaikka äidinkieli olisi molemmille lähestulkoon sama.

Kulttuurien välillä on myös eroja miten asiat esitetään. Viestinnän ja kulttuurin välinen suhde ilmenee ainakin sen perusteella, onko henkilö matalan kontekstin kulttuurin maasta vai korkean kontekstin kulttuurin maasta. Matalan kontekstin kulttuurin maassa, kuten Hollanti, asiat esitetään seikkaperäisesti sanallisessa muodossa, hyvin selkeästi ja suoraan. Erran ja Tjia [2005, 184] esimerkissä kuvattiin miten intialainen ohjelmoija (nainen) oli juuri aloittanut työt hollantilaiselle asiakkaalle. Kyseinen nainen löydettiin eräänä aamuna itkemästä koneensa ääreltä. Hollantilainen projektipäällikkö oli lähettänyt hänelle sähköpostia, jossa hän oli suoraviivaisesti listannut asioita, joissa nainen oli tehnyt

virheitä. Nainen piti viestiä loukkaavana. Hollantilaisen projektipäällikön ei ollut tarkoitus viestillään loukata naista henkilökohtaisesti, ainoastaan listata asiat mitkä vaativat naiselta toimenpiteitä.

## 2.5 Transaktiivinen muisti, kollektiivinen älykkyys ja ryhmätietoisuus

Transaktiivisen muistin on määritellyt alun perin määritellyt Daniel Wegner [1987] vastinaena aiemmille teorioille kuten ryhmäajattelu. Wegner [1987] määritteli transaktiivisen muistin seuraavasti: Transaktiivinen muisti on joukko yksilöiden muistijärjestelmiä ja niiden välillä tapahtuvaa viestintää. Oshri [2008] on tutkinut transaktiivisen muistin vaikutusta tietämyksen siirtämiseen hajautetussa ryhmässä. Nykyaikainen trendi töiden ulkoistamiselle kokonaisuuksina, osakokonaisuuksina tai jotain siltä väliltä luo tarpeen tietämyksen siirtämiselle kahden tai useamman toisistaan etäällä olevan ryhmän välillä. Tietämyksen ja osaamisen siirtäminen erityisesti toisistaan etäällä olevien ryhmien välillä ei ole ongelmatonta. Oshrin [2008] tutkimus selvittää, miten transaktiivinen muisti vaikuttaa tietämyksen siirtämiseen fyysisesti etäällä olevien ryhmien ja henkilöiden välillä. Yhtenä erityistapauksena hajautetusta ryhmästä voidaan pitää virtuaalista tiimiä. Virtuaalitiimin koostumus saattaa olla sen kaltainen, että jokainen virtuaalitiimin jäsen on fyysisesti etäällä ryhmän muista jäsenistä. Virtuaalitiimiä käsitellään tarkemmin kohdassa 2.8.

Etäisyyden ja aikaerojen negatiivisten vaikutusten lieventämiseksi käytetään erilaisia teknologisia ratkaisuja sekä tarpeen mukaan kasvokkain käytävää viestintään. Kuitenkin kokemukset osoittavat, että pelkkä teknologia ei välttämättä riitä poistamaan ongelmia tietämyksen siirtämisessä etäällä olevien ryhmien välillä. Tästä syystä on tutkittu mahdollisuutta käyttää transaktiivista muistia ryhmien välisessä tietämyksen siirtämisessä. Seuraavassa on lyhyt kuvaus Wegnerin [1987] tutkiman ja määrittelemän transaktiivisen muistin prosessin eri osista. Pääpiirteittäin prosessi koostuu kolmesta eri osasta: tietämyksen koodaamisesta, tietämyksen tallentamisesta sekä tietämyksen hakemisesta. Prosessin lopputuloksena pitäisi olla tietämyksen siirtyminen henkilöltä toiselle.

Tietämyksen hallintaan liittyvä kirjallisuus olettaa yleisesti, että tietämyksen tyyppi (täsmällinen ja hiljainen tieto) määrittelee tavan miten tietämys koodataan, tallennetaan ja haetaan (luokiteltu ja henkilökohtainen muistijärjestelmä).

Täsmällisellä tiedolla tarkoitetaan esimerkiksi dokumentaatiota kuten vaatimusmäärittely tai vastaava, joka koodataan, tallennetaan luokiteltuun muistijärjestelmään sekä jatkossa haetaan luokitellusta muistijärjestelmästä. Täsmällisen tiedon koodaamisella tarkoitetaan

dokumentaation sisällön määrämuotoista esitystapaa, jossa sanasto, käsitteet jne. ovat määritelty niin, että jokainen dokumentin käyttäjä (kirjoittaja/päivittäjä/lukija) ymmärtää asiat yhdenmukaisesti.

*Hiljaisella tiedolla* (tacit knowledge) tarkoitetaan tietämystä, jota henkilö on ajan ja kokemuksen myötä kerryttänyt henkilökohtaiseen muistijärjestelmään. Hiljaisen tiedon koodaaminen tarkoittaa sitä, kun henkilö itse tietää jotain tai tietää, että joku muu tietää jotain ja tieto siitä mitä joku tietää tallennetaan ja kategorisoidaan omaan muistijärjestelmään, josta se voidaan tarvittaessa hakea. Hiljaisen tiedon siirtäminen vaatii aina kasvokkain tai ainakin henkilökohtaisesti tapahtuvaa viestintää henkilöltä henkilölle tai henkilöltä ryhmälle.

Transaktiivisen muistin hyödyiksi voidaan katsoa sen antama mahdollisuus henkilöiden erikoistua omalla osa-alueella erityisen eteviksi. Tämän mahdollistaa tieto siitä, että henkilö tietää mitä muut tietävät ja näin ollen hänen ei henkilökohtaisesti tarvitse omaksua kyseistä tietoa tarkemmin, riittää että hän tietää kuka tietää ja miten kyseiseen henkilöön saa yhteyden. Tästä seuraa mahdollisuus muun muassa siihen, että ryhmän sisällä tietämys syvenee ajan myötä ja näin ollen koko ryhmä hyötyy siitä. Ulospäin näkyviä hyötyjä ovat esimerkiksi ryhmän tehokkuuden paraneminen sekä tuotteet saadaan nopeammin markkinoilla.

Transaktiivisen muistin hyödyntäminen esimerkiksi hajautetussa ohjelmistotuotannossa ei ole niin yksioikoista, kuin se todennäköisesti on projektissa, jossa ryhmä tai ryhmät työskentelevät samoissa tiloissa. Vaihtuvuus projektin ryhmien jäsenistössä vaikuttaa merkittävästi transaktiivisen muistin kehittymiseen, koska transaktiivisen muistin käyttöönotto ja sen kehittäminen ryhmässä on aikaa vievä prosessi. Jos vaihtuvuus ryhmässä on suurta, ei transaktiivisen muistin käyttöönotto ole ehkä mahdollista tai edes mielekästä. Vaihtuvuuden lisäksi on myös todennäköistä, että eri ryhmien jäsenet eivät ole aiemmin työskennelleet yhdessä. Näin ollen ryhmien kesken ei ole yhteistä kokemuspohjaa sekä luottamusta, joka syntyy tai on syntymättä henkilöiden välille ajan myötä ja yhteisten kokemusten pohjalta.

Kollektiivisella älykkyydellä tarkoitetaan sellaista toimintaa tai älykkyyttä, jota ilmenee enemmän yhteisön keskuudessa kuin yksittäisten ihmisten keskuudessa. Kollektiivisen älykkyyden yksi ilmentymä on edellä mainittu transaktiivinen muisti [Hakkarainen, 2003]. Hakkaraisen [2003] mukaan ihmisellä kollektiivista älykkyyttä ilmenee sekä vertauskuvainnollisesti että käytännön yhteisöllisinä älyllisinä prosesseina, joita voidaan verrata parviälyyn. Esimerkkeinä yhteisöllisestä kollektiivisesta älykkyydestä Hakkarainen [2003] mainitsee mm. *sosiaalisen navigaation* sekä *episteemisen navigaation*.

Ensimmäisellä tarkoitetaan miten muiden ihmisten toimet vaikuttavat ihmisen tekemiin valintoihin ja tekoihin. Esimerkkinä on mainittu muun muassa, miten ihminen valitsee ravintolan vieraassa paikassa. Todennäköisesti hän valitsee mieluummin sellaisen ravintolan, jossa on muita ihmisiä kuin sellaisen ravintolan, joka on hiljainen. Samoin ihmisellä on taipumus ottaa huomioon esimerkiksi netistä ostaessa, mitä muut samoista asioista kiinnostuneet ihmiset ovat ostaneet. *Episteemisellä navigaatiolla* tarkoitetaan eräänlaista teoreettista tutkimusta. Uuden tiedon luominen vaatii yleensä sitä, että ensin opitaan ymmärtämään millaisia tuloksia muut ihmiset ovat tieteenalalta jo tuottaneet ja näiden tietojen pohjalta on helpompaa edetä ja raivata omaa polkua eteenpäin kyseisen tieteenalan viidakossa. Usein kollektiivisen älykkyyden aiheuttamat toimet ovat enemmänkin ihmisen itseorganisoidumisen ansiota kuin, että sen takana olisi jokin tietty yhteisö tai organisaatio. Tosin voisi kuvitella, että nykyään sosiaalisen median ansioista tai syystä erilaiset ilmiöt saattavat syntyä yksittäisenkin ihmisen aloitteesta. Tosin tämän kaltaiset ilmiöt ovat usein kovin lyhytkestoisia. Kollektiivisen älykkyyden tutkimuksen perusoletta on, että kaikki vähänkään merkittävät, älykkäät saavutukset perustuvat kulttuurien ja yhteisöiden tekemiin havaintoihin, kokemuksiin ja tietoon. Ihmiselle on luontaista tukeutua erilaisiin apuvälineisiin helpottaakseen omaa oppimisen taakkaa. Näistä Hakkarainen [2003] mainitsee muun muassa helmitaulun, laskutikun, laskukoneen ja tietokoneen. Erilaisten apuvälineiden lisäksi esimerkiksi asiantuntijuuden kasvattaminen ei ole pelkästään muutos kognitiivisissa prosesseissa vaan kyseisen asiantuntijakulttuurin kielen omaksumisesta ja taidosta käyttää sitä.

## 2.6 Tietämyksenhallinta ja tiedon jakaminen

Tässä kohdassa käydään läpi mitä tarkoitetaan seuraavilla käsitteillä: *tietämyksenhallinta* (knowledge management), *tietämyksen jakaminen* (knowledge sharing), *jaettu tietämys* (shared knowledge) sekä *yhteisymmärrys* (shared understanding) ja mitä ne tarkoittavat erityisesti hajautetussa ohjelmistosuunnittelussa.

Tietämyksenhallinta on menetelmä, joka yksinkertaistaa prosessia, jolla organisaatio jakaa, levittää, luo, tallentaa sekä ymmärtää organisaation oman tietämyksensä [Boden *et al.*, 2009]. Kulttuurierojen (kansallinen, organisaation, ammatillinen) vaikutuksesta tietämyksenhallintaan on tutkinut mm. Boden [2009] ja todennut tietämyksenhallinnasta seuraavaa: Tietämystä on kirjallisuudessa tulkittu ikään kuin tuotteena, jossa siitä voidaan irrottaa paikkaan ja tilanteeseen liittyvät yhteydet ja näin ollen tieto voidaan tallentaa vaikkapa tietokantaan jälleen jaettavaksi. Usein näin ei kuitenkaan ole, vaan tietämys on sidottu tiukasti johonkin asiayhteyteen. Tämän tyyppisestä tietämyksestä hyvä esimerkki on hiljainen tieto. Hiljaista tietoa ei voida suoraan tallentaa tietokantaan ilman, että siitä menetetään osa tiedosta. Menetelmää ryhmän hiljaisen tiedon määrän mittaamiseksi ovat

tutkineet Ryan ja muut [2009]. Hiljaiseen tietoon voidaan tutkimuksen mukaan liittää kolme ominaispiirrettä:

1. hiljaisen tiedon hankkimiseen ei ole käytetty ympäristön tukea,
2. hiljainen tieto on enemmänkin tietoa miten asia tehdään kuin tietoa itse asiasta,
3. hiljainen tieto on käytännöllistä tietoa, jolla voidaan saavuttaa henkilökohtaiset tavoitteet.

Ryhmän hiljainen tieto taas on määritelty seuraavasti: ryhmän hiljainen tieto on kokoelma hiljaista, yksilöllistä, tavoitehakuista ammatillista tietoa, johon eri ryhmän jäsenet tarjoavat oman näkökulmansa [Ryan et al., 2009]. Liittyen yhteistyöhön ja koordinointiin Milewski [Boden and Avram, 2009] on esittänyt käsitteen ”bridges” sosiaalisen verkoston näkökulmasta. Käsitteen ”bridges” voisi suomentaa vaikkapa ”sillanrakentajaksi”. Tutkimuksen mukaan ihminen on keskeisessä asemassa sosiaalisissa verkostoissa. Tämän tyyppiselle henkilölle on kirjallisuudessa useampikin nimitys, kuten ”information broker”, ”boundary spanner”, ”gatekeeper” tai ”cultural liaison”. Kaikilla näillä käsitteillä on enemmän tai vähemmän sama merkitys. Hajautetussa projektissa osa ihmisistä toimii luontaisesti ”sillanrakentajan” roolissa ja ottaa enemmän vastuuta tietämyksen jakamisesta kuin muut [Boden and Avram, 2009]. Tämä asia on hyvä tiedostaa, kun projekti halutaan esimerkiksi hajauttaa useammalle tiimille, useampaan kuin yhteen paikkaan ja jäseniä valitaan tehtäviin. Jokaisessa projektissa olisi hyvä olla henkilö, jonka sosiaaliset taidot loisivat mahdollisimman hyvän pohjan tiimien väliselle vuorovaikutukselle. Erran ja Tjia [2005] ovat kirjassaan todenneet oikean henkilön valitsemisesta hajautettuun ohjelmistoprojektiin seuraavasti: ”Henkilöt jotka osaavat luoda sosiaalisia suhteita kasvokkain, ovat myös usein niitä jotka osaavat luoda sosiaalisia suhteita verkossa”. Jossain tapauksessa tiimin asiantuntija toimii luontevasti ”sillanrakentajan” roolissa, koska hänellä on yleensä hyvä ja syvä tietämys järjestelmän toiminnasta ja siten mahdollisuus jakaa ja vastaanottaa tietämystä muiden tiimien välillä. Toisaalta tiimin vetäjät tai projektipäällikkö voivat toimia myös luontevasti ”sillanrakentajan” roolissa, koska he ovat joka tapauksessa vastuussa ryhmien toiminnasta ja sitä myöden myös vastuussa ryhmien välisten asioiden hoidosta.

Tietämyksen jakamisesta virtuaalitiimissä on käsitelty mm. [Hinds and Weisband, 2003]. Erityisesti virtuaalitiimin maantieteellinen hajautuneisuus aiheuttaa haasteita tietämyksen jakamisella. Virtuaalitiimin viestintä tukeutuu vahvasti erilaisiin viestintävälineisiin kuten sähköpostiin, puhelimeen, pikaviestimiin jne. Tämä vaikeuttaa tietämyksen jakamista sekä yhteisen tietämyspohjan rakentamista. Jos hajautetulla ryhmällä, kuten virtuaalitiimillä ei ole yhteistä tietämyspohjaa, niin sillä saattaa olla vaikutusta ryhmän suorituskykyyn. Tämä johtuu siitä, että ilman yhteistä tietämyspohjaa yksittäiset ihmiset virtuaalitiimissä



eivät osaa ennakoida, miten toiset tulevat suoriutumaan omista tehtävistään. Tämä taas voi aiheuttaa ylimääräistä tarvetta tarkkailla toisten tekemisiä, joka aiheuttaa mahdollisesti töiden tekemistä useamman henkilön toimesta ja sillä on suora vaikutus ryhmän suoriutuskykyyn. Tällöin taas luottamus ryhmän muihin jäseniin huononee sekä edesauttaa turhautumista toisiin ryhmänjäseniin. Kun ryhmän sisällä vallitsee hyvä yhteisymmärrys siitä mitä pitää tehdä, sekä luottamus toisiin ryhmänjäseniin, mahdollistaa se ryhmälle asetettujen tavoitteiden saavuttamisen ja se on omiaan lisäämään ryhmän jäsenten motivaatiota saavuttaa yhteinen tavoite. Hinds ja Weisband, [2003] on määritellyt viisi hyötyä siitä, että ryhmällä on yhteisymmärrys, jaettu näkemys ryhmän sisällä sekä mitä ollaan tekemässä ja miksi:

- Mahdollistaa ryhmän jäsenten ennakoida muiden ryhmän jäsenten käyttäytymistä.
- Mahdollistaa resurssien tehokkaan käytön.
- Ehkäisee ongelmia sekä virheitä toteutuksessa.
- Lisää ryhmän jäsenten tyytyväisyyttä ja motivaatiota.
- Vähentää ryhmän jäsenten turhautumista sekä ristiriitoja ryhmän jäsenten välillä.

Hinds ja Weisband, [2003] tarjoaa seuraavia mahdollisuuksia yhteisymmärryksen vahvistamiseksi virtuaalisessa tiimissä: Ryhmän jäsenten tulisi oppia *tuntemaan toisensa, samaistua toinen toisiinsa ja vahvistaa yhteisiä odotuksia toisiaan kohtaan*. Yhteisymmärryksen ulottuvuuksia Hinds ja Weisband, [2003] on luetellut seuraavasti:

- Ryhmän tavoitteiden luonne.
- Ryhmän työt ja tehtävät.
- Ryhmän käyttämät prosessit töistä suoriutumiseen.
- Vuorovaikutus ryhmän sisällä tehtävien tueksi.
- Tietoa muiden ryhmänjäsenten tekemisistä ja toimista.

Hinds ja Weisband, [2003] on listannut asioita, jotka myötävaikuttavat tiimin jäsenten välisen yhteisymmärryksen syntymiseen: *ryhmän jäsenten samankaltainen tausta, ryhmän jäsenten yhteiset kokemukset, Mahdollisuus tutustua ajan saatossa paremmin toiseen, viestintä ja tiedonjakaminen sekä ryhmähengen kehittäminen*.

Granovetter [Boden *et al.*, 2009] on painottanut sosiaalisen yhteyksien roolia organisaation toimintoina. Samaten käsitteellä sosiaalinen pääomalla viitataan ominaisuuksiin kuten molemminpuolinen tuki, yhteinen kieli, jaetut normit, sosiaalinen luottamus ja yhteinen sitoutuneisuus. Tämä voidaan ymmärtää eräänlaisena sosiaalisena "liimana", joka pitää yhteisöt kasassa ja on korostettu vapaaehtoisuuden luonnetta yhteisössä sekä sen tukea organisaation tietämyksenhallintaan [Boden and Avram, 2009]. Ryhmän jäsenten

samankaltainen tausta voi tarkoittaa esimerkiksi sitä, että he ovat opiskelleet samassa oppilaitoksessa, työskennelleet samassa työpaikassa, heillä on yhteisiä ystäviä organisaatiossa tai työn ulkopuolelle, heillä on samankaltainen sosiaalinen tausta muutamia mainitakseni. Samankaltainen tausta mahdollistaa yhteisen kielen ja käsitteistön ymmärtämisen ja luottamuksen synnyttäminen toiseen saattaa käydä helpommin. Yhteinen tausta voi myös vaikuttaa myönteisesti ja mahdollistaa ryhmän jäsenten välisen luontevamman viestinnän, koska heiltä löytyy mahdollisesti yhteisiä kiinnostuksen kohteita tai epäkohtia, joita ovat taustansa takia joutuneet kokemaan.

Yhteiset kokemukset voivat olla peräisen kaukaakin menneisyydestä, kuten lapsuudesta, koulusta, opiskelua ajoilta, autokoulusta, harrastusten parista, armeijasta, samasta työpaikasta ja niin edelleen, listaa voisi jatkaa lähes loputtomiin. Yhteisillä kokemuksilla on vaikutusta ainakin ryhmähengen paranemiseen, viestinnän helpottumiseen, luottamuksen lisääntymiseen jne. Voidaan olettaa, että yhteiset kokemukset, joissa lopputulema on ollut positiivinen vaikkakin itse kokemus ei olisi ollut miellyttävä, vahvistaa edellä mainittuja asioita. Toisaalta yhteisillä kokemuksilla saattaa olla myös ristiriitainen vaikutus, jos joku kokee sen negatiivisesti ja toiset taas positiivisesti. Hyvänä esimerkkinä voisi olla vaikkapa organisaation järjestämä yhteinen hengennostatus tilaisuus, jonka joku kokee itseään nöyryyttävänä tai loukkaavana ja muut taas kokevat tilanteen miellyttävänä, sellaisena kuin se todennäköisesti tarkoitettiin.

Toisen tunteminen entuudestaan ja esimerkiksi työskentely pitkään samassa organisaatiossa mahdollistavat henkilöiden välisen suhteen syventyä ajan myötä. Tässäkin tapauksessa on mahdollista myös käydä päinvastoin, henkilöiden suhteet voivat tulehtua ajan myötä erittäin huonoiksi. Suhteen syveneminen helpottaa yhteisymmärryksen syntymistä henkilöiden välille, kun taas suhteen tulehtuminen aiheuttaa helpommin ristiriitatilanteita. Esimerkkinä voidaan ottaa vaikkapa tilanne, jossa henkilöt ovat työskennelleet aiemmin usean vuoden samoissa tehtävissä ja tämän jälkeen he ovat siirtyneet omille urilleen. Jos henkilöiden välinen suhde on ollut toimiva, niin vuosien jälkeenkin näiden henkilöiden ajautuessa esimerkiksi samaan projektiin, yhteisymmärryksen muodostaminen on huomattavasti luontevampaa, kuin jos henkilöt eivät olisi tunteneet toisiaan aiemmin. Toisaalta, jos henkilöiden välinen suhde on ollut kaikkea muuta kuin toimiva, niin yhteisymmärryksen ja luottamuksen muodostaminen on jopa vaikeampaa kuin jos henkilöt eivät ennestään tuntisi toisiaan.

Tiedon jakaminen ja viestintä mahdollistavat yhteisymmärryksen vahvistumisen. Ongelmien ratkominen yhdessä ja omien mielipiteiden jakaminen sekä toisten mielipiteiden kuunteleminen vahvistaa yhteisymmärrystä.

Hyvä ryhmähenki mahdollistaa myös ryhmän yhteisymmärryksen paranemisen. Tärkeätä on kuitenkin muistaa, että ryhmähenkeä kohottaessa on pidettävä huolta siitä, että koko ryhmä on mukana, eikä ryhmässä tapahtuisi syrjäytymistä.

## 2.7 Ryhmän suorituskkyky

Miten määritellään ryhmän suorituskkyky ja millä asioissa on vaikutusta suorituskkykyyn?

Eräs määritelmä ryhmän suorituskkyvylle:

”Ryhmän suorituskkyky on tulosta ryhmän jäsenten välisestä vuorovaikutuksesta sekä ryhmädynamiikasta” [Guinan, Coopridge and Faraj, 1998].

Empiirisen analyysin mukaan ryhmän hajauttaminen maantieteellisesti vaikuttaa negatiivisesti tuottavuuteen sekä laatuun. Hajauttamisesta aiheutuvia ongelmia voidaan vähentää ottamalla käyttöön ohjelmistotuotantoprosesseja [Narayan and Balan, 2007].

Hyvin järjestetty tietämyksenhallinta on tärkeä osa onnistunutta ohjelmisto projektia, vaikuttaen ohjelmiston laatuun sekä ryhmän suorituskkykyyn [Boden *et al.*, 2009].

Ryan ja muut, [2009] ovat tutkimuksessaan yrittäneet kehittää tavan mitata hiljaista tietoa ryhmän tasolla. Tutkimuksen mukaan siinä esitelty mittautapa osoittautui validiksi ja luotettavaksi, mutta merkittävää näyttöä siitä, että hiljaisen tiedon tasosta ryhmätasolla voisi vetää suoria johtopäätöksiä ryhmän tehokkuuteen. Sen sijaan löytyi näyttöä siitä, että hyvässä vuorovaikutuksessa olevalla ryhmällä on mitattavissa laajempi hiljainen tieto. [Ryan *et al.*, 2009].

Eräässä tutkimuksessa selvitettiin miten ryhmän tuttuus ja työtehtävien tuttuus vaikuttavat ryhmän suorituskkykyyn. keskeisimmät käsitteet tutkimuksessa olivat: *Muutosvaatimus* (MR), *suorituskkyky* (performance), *ryhmän tuttuus* (team familiarity), *tehtävän koko* (task size), *kompleksisuus* (structural complexity), *ryhmän koko* (team size) ja *maantieteellinen hajautuneisuus* (geographical dispersion) [Espinosa *et al.*, 2007].

Muutosvaatimuksella tarkoitetaan joko uuden ominaisuuden toteuttamista tai olemassa olevan ominaisuuden muokkaamista. Suorituskkyvylle tarkoitetaan aikaa, jossa saadaan tehtyä muutospyyntö virheettömästi. Ryhmän tuttuudella tarkoitetaan sitä kuinka monen muutospyyntöön kanssa pari on työskennellyt yhdessä pois lukien nykyinen muutospyyntö. Ryhmän tuttuus on keskiarvo ryhmän kaikille pareille. Tehtävän koolla tarkoitetaan muutospyyntöjen lisätyjen, poistettujen ja muokattujen koodirivien määrä tuhansissa. Kompleksisuudella tarkoitetaan muutospyyntöjen muokattavien

moduulien määrä. Ryhmän koolla tarkoitetaan kehittävien määrä, jotka tekivät muutosvaatimuksen vaatimat muutokset, lisäykset tai poistot. Maantieteellisellä hajaantuneisuudella tarkoitetaan miten ryhmän jäsenet sijoittuvat maantieteellisesti toisiinsa nähden [Espinosa *et al.*, 2007].

Todettiin, että tehtävän tuttuuden vaikutus laskee, kun tehtävä on kompleksisempi eli mitä kompleksisempi tehtävä, sitä vähemmän tehtävän tuttuudella on positiivista vaikutusta. Tehtävän tuttuuden todettiin olevan riippumaton tehtävän koosta eli tehtävän tuttuudesta saatu hyöty ei ole riippuvainen tehtävän koosta. [Espinosa *et al.*, 2007].

Ryhmän tuttuuden todettiin olevat hyödyksi, kun ryhmän koordinointi on haasteellisempaa. Toisin sanoen silloin, kun ryhmän koko kasvaa tai ryhmä on maantieteellisesti hajaantunut [Espinosa *et al.*, 2007].

Tutkimuksessa tuttuuksien (tehtävä ja ryhmä) välillä havaittiin yllättävä seikka. Olettamuksista poiketen tehtävä tuttuuden ja ryhmä tuttuuden katsottiin olevan enemmän toistensa poissulkevia kuin täydentäviä ominaisuuksia. Tehtävän tuttuus siis lisää ryhmän suorituskkyä enemmän silloin, kun ryhmän tuttuus on pienempi ja päinvastoin [Espinosa *et al.*, 2007].

## 2.8 Virtuaalitiimi

Tässä kohdassa avataan hieman käsitettä virtuaalitiimi. Yhdyssana muodostuu kahdesta sanasta virtuaali ja tiimi. Molemmat sanat ovat ajan hengessä lainasanoja englannin kielen sanoista virtual ja team.

Sanan virtuaalinen etymologia kertoo sanan pohjautuvan latinan kielen sanaan virtus, jolla tarkoitetaan yliveraisuutta, kykyä ja tehokkuutta. Sana näin ollen ei täysin kuvaa tämän päivän käyttötarkoitusta. Sana "virtual" kääntyy suomenkieleen muun muassa seuraavasti: vale, näennäinen, virtuaalinen tai näennäis-.

Mikä taas erottaa sanan tiimi esimerkiksi sanoista ryhmä tai joukko. Seuraavassa käydään läpi miten tiimi eroaa satunnaisesta ryhmästä tai joukosta. Nykyään on hyvin selkeä käsitys siitä, mitkä ominaisuudet kuvaavat pientä ryhmää. Seuraavat ominaispiirteet kuvaavat pientä ryhmää yleisesti [Lipnack and Stamys 2000]:

- Kaksi tai useampi yksilö.
- Ryhmän jäsenten välinen vuorovaikutus.
- Keskinäinen riippuvuus

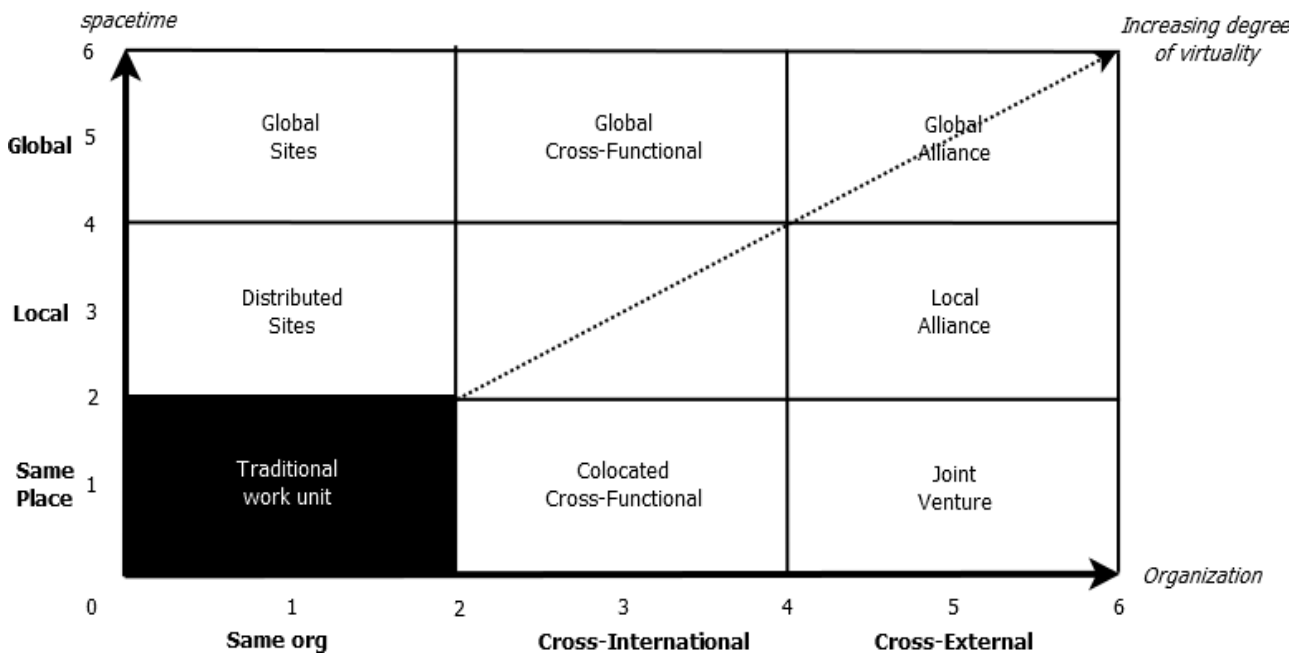
Näihin pohjautuen Lipnack ja Stamps [2000] ovat esittäneet yksinkertaisen määritelmän kuvaamaan pientä ryhmää: "Individuals interacting interdependently". Tämän voi suomentaa vapaasti vaikkapa seuraavasti: "Toisistaan riippuvaiset yksilöt ja heidän välinen vuorovaikutus". Yhteinen etu muodostaa ihmisistä ryhmän, tämä erottaa ryhmän satunnaisesta ryhmästä ihmisiä eli joukosta. Lipnack ja Stamps [2000] on määritellyt ryhmän muodostuksen seuraavasti: "Joukosta ihmisiä tulee ryhmä, kun kokonaisuus on enemmän kuin osiensa summa." Harvemmin ryhmä muodostuu ilman erityistä tarvetta. Miten ryhmänjäsen sitten tuntee olevansa osa jotakin ryhmää? Esimerkkinä voisi mainita vaikkapa jääkiekkjoukkueen. Ryhmä ei muodostu tyhjästä vaan tiettyä tarkoitusta varten, tässä tapauksessa pelatakseen yhdessä jääkiekkoa. Ryhmän jäsenillä on pääsy esimerkiksi joukkueen yhteisiin tiloihin ja mahdollisuus saada tietoonsa vain joukkueelle kuuluvia asioita. Näin voidaan määritellä henkilön joko kuuluvan joukkueeseen tai sitten ei. Sama pätee myös työyhteisössä, jossa esimerkiksi tietty tieto kuuluu vain jollekin tietylle ryhmälle, eikä siihen kuulumattomille jäsenille. Ryhmäytymistä saattaa haitata se, että osa ryhmän jäsenistä ei kuulu samaan organisaatioon. Tämä voi johtua esimerkiksi siitä, että osa ryhmästä työskentelee alihankkijana ja näin ollen ei ole oikeutettu pääsemään käsiksi kaikkeen tietoon, mihin muulla ryhmällä on oikeus. Yhteinen päämäärä aiheuttaa tarpeen ryhmän jäsenten väliselle vuorovaikutukselle. Keskinäinen riippuvuus jäsenten välillä juontaa juurensa ryhmän yhteisistä tavoitteista sekä ryhmän yksilöiden erityisistä osaamisalueista, joista muut ryhmän jäsenet voivat hyötyä. [Lipnack and Stamps, 2000, s.55-57]

Mikä sitten on tiimi? Kirjallisuus ilmaisee asian yksiselitteisesti seuraavasti: Tiimi on olemassa jotain tiettyä tehtävää varten. Tiimin muista ryhmistä erottaa siis sen tehtävä orientoitunut luonne. Vaikka jokainen ryhmä suorittaa jonkinlaisia tehtäviä, on tehtävä tiimin pääasiallinen huomion kohde. Tarkoitus on jokaiselle ryhmälle elinehto, mutta tiimi on erityisesti, tarkoituksen mukaisesti ja poikkeuksetta olemassa lopputulosta varten. Tehtävä on jokin työ, mihin tiimi yhdessä sitoutuu ja tekee töitä, jotta projektin päätteeksi tehtävät saadaan tehtyä. Tiimin jäsenet valikoituvat myös usein sen perusteella millaisia erityistaitoja tehtävän suorittaminen vaatii. Koska tehtäväkeskeisyys siis erottaa tiimin muista ryhmistä, niin perinteisen ja virtuaalitiimin erottaa virtuaalitiimin tarve ylittää erilaisia raja-aitoja kuten välimatka, aikaerot, kulttuuriset erot ja kielierot [Lipnack and Stamps, 2000, s.55-57].

Paasivaara [2005] on avannut virtuaaliorganisaatio käsitettä seuraavasti. Virtuaaliorganisaatio käsitteellä on monta määrittelyä, mutta seuraavat asiat ovat kaikille yhteisiä. Virtuaaliorganisaatio on yleensä maantieteellisesti sekä mahdollisesti kulttuurisesti hajautunut, fyysisen etäisyyden, aikaeron ja organisaatorajat ylittävä

toiminnallinen yksikkö. Rakenteeltaan virtuaaliorganisaatiot eivät ole hierarkkisia, eikä valta ole keskittynyttä. Luonteeltaan virtuaaliorganisaatiot ovat väliaikaisia, johonkin tehtävään perustettuja ketteriä yksiköitä, joita on helppo mukauttaa aina tarpeen mukaan. Kun virtuaaliorganisaatiolle asetettu tehtävä on täytetty se lakkaa olemasta.

Lipnack ja Stamps [2000] on esittänyt virtuaaliuden astetta kuvan 1 mukaisesti.



Kuva 1. Organisaation virtuaalisuuden taso.

Kuvassa x-akselilla kuvataan virtuaalitiimin organisaation tyyppiä. Lähimpänä origoa on perinteinen yksikkö, jossa kaikki työskentelevät samassa organisaatiossa (Same org). Toisessa äärilaidassa organisaatio on hajautunut (Cross-External).

Kuvassa y-akseli kuvaa paikkaa ja aikaa. Lähimpänä origoa kaikki työskentelevät samassa paikassa (Same Place) ja toisessa ääripäässä työtä tehdään globaalisti (Global)

Esimerkiksi jos virtuaalitiimin jäsenet kaikki työskentelevät samassa organisaatiossa (x-akseli) sekä samassa paikassa (y-akseli), on tämän virtuaalisuuden aste kaikkein pienin. Akselien toisissa ääripäässä organisaatio toimii paikan ja ajan suhteen täysin globaalisti, tällöin organisaatio on kansainvälinen yhteenliittymä (Global Alliance).

## 2.9 Joukkoistaminen

Howe [Estellés-Arolas and González-Ladrón-De-Guevara, 2012] on esittänyt ajatuksen joukkoistamisesta vuonna 2008. Ajatuksen taustalla on teettää työ suurella joukolla

määrittelemättömiä ihmisiä, joita yhdistää internet. Vastaavanlaisen ajatuksen ”group sourcing” on määritellyt Chamberlain [2014]. Tämän ”ryhmyttämisen” perusajatus on seuraava: suorittaa tehtävä ryhmällä olennaisesti kiinnostuneita ihmisiä laajalla osaamistaustoilla, joita yhdistää sosiaalinen media.

Olisiko mahdollista siirtää group sourcing ajatusmallia soveltuvien osien tukemaan laajan järjestelmän testaamista, jossa järjestelmän tekeminen ja testaaminen on hajautettu maantieteellisesti useaan paikkaan ja näin ollen järjestelmäosaaminen järjestelmän eri osa-alueista on myös pirstoutunut ympäri maailmaa. Esimerkiksi viestinnän tueksi sekä järjestelmätuntemuksen levittämiseksi saattaisi löytyä hyviä ideoita kuten sosiaalinen media, joka tässä tapauksessa on rajoitettu tietyn ryhmän käyttöön. Perusajatuksena olisi se, että organisaation sisäisessä sosiaalisessa mediassa eri vaiheiden testaamiseen (yksikkötestaus, integrointitestaus, järjestelmätestaus jne.) liittyviä ongelmia tuotaisiin toisten käyttäjien ratkottavaksi. Näin ongelma saataisiin mahdollisimman laajalle käyttäjäryhmälle, jolla on kokemusta järjestelmän eri osaamisalueilta. Käyttäjät voivat olla ketä tahansa organisaation henkilöstöä, jotka ovat kiinnostuneita koko järjestelmän toiminnasta sekä oman osa-alueensa tietämystä. Tämä olisi myös todennäköisesti hyvä kanava jakaa erilaisia hyväksi havaittuja käytäntöjä ja esimerkkejä virheistä, jotka ovat päässeet aina asiakkaalle asti.

## 2.10 Ohjelmiston laatu

Kuinka ohjelmiston laatua voidaan parantaa, erityisesti vikojen pääsemistä asiakkaalle asti? Mitä aiemmin ohjelmistossa olevat viat havaitaan ja ne voidaan korjata, sitä parempi. Mitä pidemmälle viat pääsevät testausprosessissa, sitä kalliimmaksi ne tulevat ja ovat aina haastavampi korjata, koska viat saattavat vaatia laajempia korjauksia tai vian aiheuttajaa voi olla haastavaa paikallistaa ohjelmistosta [Patton, 2001].

Mahdollisia syitä siihen, miksi testaaminen saattaa joskus jäädä ilman riittävää huomiota, löytyy muun muassa seuraavista seikoista: tiukka budjetti, haasteellinen aikataulu sekä resurssipula ovat usein päällimmäisiä syitä. Brooks [1975] arvioi, että optimismi on syynä testauksen liian vähäiseen huomioimiseen aikataulusvaiheessa. Vikojen määrän oletetaan olevan pienempi kuin se todellisuudessa osoittautuu. Brooks [1975] havaintojen mukaan harvempi varaa aikatauluissa puolet testaamiselle, mutta todellisuudessa käyttää puolet ajasta juurikin testaamiseen. Rahaa tarvitaan testauksen osalta ainakin osaaviin resursseihin, yhteen tarkoitukseen soveltuvaan testausympäristöön tai jopa useampaan, testausympäristöjen ylläpitoon sekä ohjelmistoihin. Budjetin ollessa tiukalla, kokonaisvaltaisesta testaamisesta voi olla helppo hieman karsia. Liian tiukat aikataulut liittyvät kiinteästi tiukaksi vedettyyn budjettiin ja

täten kokonaisvaltaisesta testaamisesta tässäkin tapauksessa voi olla helppo joustaa. Kokonaisvaltaista testaamista saattaa olla vaikea perustella, jos se nostaa projektin hintaa tasolle, jolloin voi olla haastavaa saada projekteja kilpailluilla markkinoilla. Slaughterin [1998] mukaan laatua voidaan pitää tarpeettomana kustannuksena tai asiana, joka aiheuttaa ongelmia aikatauluissa. Asennetta kuvaa hyvin erään esimiehen lausunto: "I'd rather have it wrong than have it late. We can always fix it later." [Slaughter, 1998]

Sovellusalueen tietämyksellä tarkoitetaan esimerkiksi testattavan kohteen sovellusalueen erityispiirteiden tuntemista. Eräässä tutkimuksessa todetaan, että sovellusalueen tietämys on elintärkeä tekijä määriteltäessä vaatimuksia. Heikko sovellusalueen tuntemus voi tuottaa laadullisesti heikon vaatimusmäärittelyn. [Haruhiko and Motoshi, 2006] Tämä voi aiheuttaa sen, että oleellisiakin asioita jää kokonaan tai osittain testaamatta. Näihin asioihin voidaan vaikuttaa, kunhan ne vain huomioidaan jo projektin budjettia, aikataulua ja resursointia suunniteltaessa. Aina ei kuitenkaan ole helppoa perustella kokonaisvaltaisen testaamisen tarvetta, jos ei ole selkeästi esittää siitä saatavaa rahallista hyötyä [Slaughter, 1998]. Nykyään suosiossa olevat ketterät ohjelmistosuunnittelumenetelmät aiheuttavat omat haasteensa testausprosessille. Näitä haasteita ovat muun muassa se, että testaamisen pitäisi kulkea käsi kädessä toteutuksen kanssa, niin että käytössä olevan periodin jälkeen jokaista uutta tai muokattua ominaisuutta kohden löytyy kattavasti testejä. Aina ei kuitenkaan testaamiselle jää riittävästi aikaa, koska esimerkiksi testausympäristö ei ole välttämättä käyttökunnossa tai ohjelmisto on testattavissa liian myöhään.

Katselmointi on yksi tärkeä laadunvarmistus menetelmä [Bacchelli and Bird, 2013]. Kun mietitään miten koodissa olevia vikoja voitaisiin vähentää mahdollisimman aikaisessa vaiheessa, niin eräs merkittävä asia testaamisen lisäksi ovat katselmukset. Yleensä ainakin vaatimusdokumentti tuotetaan projektin alussa ja sen katselmoinnilla pyritään varmistumaan, että korkean tason vaatimukset ovat asian mukaisesti kirjattu ylös. Koodikatselmuksen tehtävänä on varmistaa uuden tai muokatun koodin oikeellisuus monessa eri mielessä. Ensinnäkin varmistetaan, että uusi tai muokattu koodi toteuttaa sille asetetut toiminnalliset vaatimukset. Vaatimukset on usein kirjattuna esimerkiksi vaatimusdokumentin muotoon ja ne pitää olla katselmoijien saatavilla jo ennen katselmointia. Katselmoinnissa on myös hyvä käydä läpi yleisiä asioita, kuten koodin ulkoasu ja tyylliseikat, jolla varmistetaan koodin helpompi ylläpito tulevaisuudessa. Näiden asioiden lisäksi on syytä katselmoida virheet, joilla ei ole suoraan vaikutusta ohjelmiston logiikan toimintaan, tämän kaltaisia virheitä ovat mahdolliset muistivuodot sekä suorituskykyyn vaikuttavat asiat. Katselmointi toimii myös erinomaisena oppimistilanteena, jossa katselmoidun koodin toimintalogiikka selkenee myös muille kuin koodin tehneelle. Katselmoinnit ovat myös hyvä tapa syventää ryhmähenkeä.



Katselmoinnissa voidaan myös tuoda esille organisaation sisällä käytössä olevia hyviä koodauskäytäntöjä ja näin ollen löytää parempia ratkaisuja toteuttaa ratkaisu. Perinteisesti koodikatselmus on ollut hyvin muodollinen sekä on vaatinut muodollisen katselmointipalaverin. Muodollisuus ja siihen liittyvät byrokraatit tekevät katselmoinnista aikaa vievän sekä joustamattoman. Bacchelli ja Bird [2013] on todennut tekemässään tutkimuksessa, että katselmoinnin muodollisuus ja synkroninen luonne vaikeuttavat sen käyttöönottoa yleismaailmallisesti. Tämä voidaan tulkita siten, että esimerkiksi hajautetussa ohjelmistokehityksessä synkroninen luonne ei välttämättä ole paras, koska muodollisen katselmoinnin järjestäminen, koodin rivi riviltä läpikäynti ei ole mielekästä esimerkiksi puhelun tai videopuhelun välityksellä. Luontevampi tapa hajautetussa ohjelmistokehityksessä on käyttää asynkronista katselmointitapaa ja sitä tukevia katselmointityökaluja. Bacchelli ja Bird [2013] nostaa esiin kuusi tärkeintä syytä miksi katselmointeja järjestetään:

1. Vikojen löytäminen.
2. Parannukset koodiin.
3. Vaihtoehtoisten ratkaisujen löytäminen.
4. Tietämyksen siirto.
5. Ryhmätietämys ja läpinäkyvyys.
6. Koodiomistajuuden jakaminen.

Vikojen löytäminen koodista tuntuu melko itsestään selvältä syytä järjestää katselmointeja. Mitä enemmän koodissa olevia virheitä havaitaan katselmoinnissa, sitä laadukkaampaa koodi on, ainakin jos laatua tulkitaan pelkästään löydettyjen virheiden valossa. Mitä myöhemmin viat havaitaan, sitä kalliimmaksi ja hankalammiksi ne tulevat korjata. Jos viat kulkeutuvat aina asiakkaalle asti, järjestelmän toimittajan maine saattaa kärsiä kolauksen, jota voi olla vaikea mitata rahassa [Slaughter, 1998].

Katselmoinnin järjestäminen koodin parantamiseksi tuntuu myös melko luontevalta. Tämä ei liity suoraan koodin toimintalogiikan korjaamiseen, vaan esimerkiksi siten on tarkoitus parantaa koodin luettavuutta ja ylläpidettävyyttä. Tämä on tärkeätä, koska yleensä koodinomistajuus ei ole sidottu yksittäiseen henkilöön vaan useampi henkilö työskentelee saman lähdekoodin parissa ja täten yhteneväinen linja koodin muotoasioissa takaa paremman luettavuuden sekä ylläpidettävyyden [Bacchelli and Bird, 2013].

Vaihtoehtoisia ratkaisuja on mahdollista löytää katselmoinnissa saatujen kommenttien perusteella. Koska jokaisella ohjelmoijalla on usein oma tapansa lähestyä ongelmia ja ratkaisut ongelmiin ovat usein toistensa kaltaisia. Tämän saattaa tarkoittaa sitä, että ratkaisut eivät aina ole ideaalisia esimerkiksi tehokkuuden, muistinkulutuksen kannalta

tai vaikkapa rakenne voi olla turhan monimutkainen suhteessa ratkaistavaan ongelmaan [Bacchelli and Bird, 2013].

Tietämyksen siirto tapahtuu melkein huomaamatta, kun katselmoijat joutuvat tutustumaan koodiin saadakseen selville tehdyn muutoksen oikeellisuuden. Pienikin muutos koodissa saattaa vaatia laajempaa tutustumista muuhun koodiin. Tämä johtuu siitä, ettei koodin toimintalogiikka välttämättä ole katselmoijalle ennestään tuttu ja varmistuakseen siitä, ettei muutos riko jo olemassa olevaa toiminnollisuutta sekä muutos itsessään on oikeellinen. Mitä laajemmalle lähdekoodiin muutos vaikuttaa, sitä työläämpää on todentaa toimintalogiikan oikeellisuus ja siinä mahdollisesti piilevät virheet [Bacchelli and Bird, 2013].

Ryhmätietämyksen (Group Awareness) käsitteen on kuvannut mm. Gutwin ja muut, [2004]. Ryhmätietämys on ymmärrys siitä, kenen kanssa työskentelet ja miten omat tekosi vaikuttavat muihin. Ryhmätietämyksestä on hyötyä tilanteissa, joissa joudutaan tekemään yhteistyötä muiden ryhmänjäsenten kanssa. Läpinäkyvyydellä kuvataan sitä, kuinka hyvin ryhmänjäseneet tietävät sen, mitä toiset tekevät. Ryhmätietämystä kuvataan hieman tarkemmin kohdassa 2.6.

Koodiomistajuuden jakaminen tarkoittaa sitä, että tietämys ja asiantuntijuus esimerkiksi jonkin ohjelmalohkon koodiin jakautuvat useammalle henkilölle. Täten vastuu koodiin tehtävistä muutoksista jakautuu useammalle henkilölle ja näin ollen esimerkiksi työn kuormitus ei kohdistu liiaksi yksittäiseen henkilöön. Tämä helpottaa myös resursointia [Bacchelli and Bird, 2013].

Pariohjelmoinnilla voidaan osaksi korvata perinteinen katselmointi. Pariohjelmoinnin kantavana ideana on sanan mukaisesti työskennellä pareina. Käytännössä koodia kirjoitettaessa, toinen kirjoittaa koodia ja toinen seuraa vierestä ja katselmoi koodia samalla ja kommentoi mahdollisesti havaitsemiaan virheitä. Myös kaksi eri tapaa lähestyä ongelmaa ja sen ratkaisua ovat tärkeä osa pariohjelmointia. Parit vaihtelevat riittävän usein paikkoja, ettei toinen pääse kyllästymään seuraamaan tai kirjoittamaan koodia. Tällä mallilla henkilöt tutustuvat väkisinkin toisiinsa ja luottamus toiseen syvenee ja se on omiaan helpottamaan kommunikointia jatkossa. Toisaalta tapauksissa, jossa molemmat henkilöt ovat verrattain kokeneita tekijöitä, ei yhteistyön sujuminen olekaan välttämättä saumatonta. Tämän taustalla voi olla se, että kokeneemmat ohjelmoivat saattavat omata tietyn tyyppiset ”omat” rutiinit ja niistä eroon pääseminen sekä muutosvastarinta vaikeuttavat uuden tyyppisen työskentelytavan omaksumista. Ainakin oppimisen kannalta parien tulisi olla valittu siten, että toinen olisi kokeneempi ja toinen nuori ja

innokas oppimaan uutta, eräänlainen mestari kisälli asetelma. [Cockburn and Williams, 2000].

Hiljaisen tiedon käsitteen esitteli Polanyi [Ryan et al., 2009], jonka mukaan hiljaista tietoa on vaikea pukea sanoiksi ja sen suosiosta huolimatta sitä on vaikea käsitteellistää ja mitata. Ohjelmiston laadun kannalta on myös merkityksellistä, miten organisaatio osaa hyödyntää oman organisaation sisällä piilevän hiljaisen tiedon ja valjastaa sen käyttöönsä sekä ottaa käyttöönsä parhaat käytännöt eri ohjelmistotuotannon vaiheissa. Tämän kaltainen osaaminen ja tietämys ovat kuitenkin tyypillisesti jatkuvasti muuttuvaa, joten käytäntöjen toimivuutta tulisi tarkkailla ja löytää mahdollisesti toimivampia käytäntöjä muuttuvien haasteiden nujertamiseksi. Omista ja muiden virheistä tulisi oppia, mutta virheiden tekemisestä ei kuitenkaan tulisi tehdä liian suurta numeroa.

Tässä kohdassa käydään läpi millaisia eri testivaiheita projektissa yleensä on sekä mitä niissä pyritään testaamaan. Alimman tason testejä on yksikkötestaus/luokkatestaus, lisäksi ohjelmistosuunnittelijalle saattaa kuulua myös moduulitestaus/prosessitestaus. Tämän lisäksi yleensä ovat seuraavat testivaiheet: integrointitestaus sekä järjestelmätestaus. Nämä testivaiheet ovat osa toiminnallista testaamista [Jenkins, 2008]. Näiden vaiheiden lisäksi on käytössä tarpeen mukaan vielä esim. suorituskykytestaus, stressitestaus sekä mahdollinen hyväksyntätestaus [Jenkins, 2008]. Seuraavassa ovat lyhyet kuvaukset eri testausvaiheista.

Yksikkötestauksella tarkoitetaan alimman tason testivaihetta, jossa ohjelmistosuunnittelija testaa tuottamansa koodin funktio/luokka -tasolla. Tällä varmistetaan yksittäisen funktion tai luokan toteutuksen oikeellisuus mm. erilaisilla syötteillä (virheelliset arvot, raja-arvot jne.) [Jenkins, 2008]. Tässä vaiheessa on helpointa saada kattavuutta testeille, koska myöhemmin erilaisten mielivaltaisten tilanteiden luominen saattaa olla hankalaa tai jopa mahdotonta. Yksikkötestaukselle on olemassa hyvä tuki mm. eri ohjelmistokehitysympäristöissä (IDE). Testit voidaan suorittaa ”nappia painamalla” ja ne on myös usein helppo automatisoida. Regressiotestauksella tarkoitetaan testien toistamista halutuin väliajoin [Patton, 2001]. Testien kattavuuden seuraamiseen on myös olemassa tuki ohjelmistokehitysympäristössä. Ketterissä menetelmissä voidaan käyttää ohjelmistosuunnittelukäytäntöä, jossa ennen uuden ominaisuuden toteuttamista tehdään ensin yksikkötestit kyseiselle ominaisuudelle ja vasta tämän jälkeen tehdään toteutus. Tätä menetelmää kutsutaan (Test Driven Development, TDD) testivetoiseksi kehitykseksi. Beck [2003] on esitellyt tämän käytännön, jossa kuvataan kolme yleistä vaihetta testivetoiselle kehitykselle:

1. Kirjoita testi.
2. Tee tarvittava koodi.
3. Tee koodista parempi.

Ensin kirjoitetaan yksikkötestit esimerkiksi funktiolle. Tämän jälkeen kaikki testit ajetaan ja todetaan, että jokainen testi todella epäonnistuu ja näin voidaan todeta, että ominaisuutta ei vielä ole toteutettu. Tämän jälkeen tehdään minimaalinen toteutus, jonka avulla saadaan jokin testeistä menemään läpi. Viimeiseksi koodia refaktoroidaan tarpeen mukaan ja siirrytään tekemään seuraavaa minimaalista toteutusta, jotta saadaan taas yksi testi menemään läpi. Näin jatketaan kunnes kaikki testit menevät läpi, koodi on refaktoroitu ja valmis toimitettavaksi [Beck, 2003].

Moduuli/prosessitestauksella tarkoitetaan yhden moduulin tai prosessin testaamista erillään järjestelmästä. Testaaminen kohdistuu pääasiassa moduulin tai prosessin tarjoamien ulkoisten rajapintojen testaamiseen. Testaamisessa on mahdollista käyttää moduulin tai prosessin käyttämien rajapintojen korvaavia toteutuksia tai käyttää oikeata toteutusta, jos se on mahdollista. Korvaavalla toteutuksella on helpompi saada aikaiseksi erilaisia virhetilanteita.

Integroititestauksessa testataan järjestelmän eri prosessien välinen toiminnollisuus. Integroititestauksessa järjestelmän käyttämät ulkoiset palvelut voidaan korvata erilaisilla korvaavilla toteutuksilla kuten simulaattoreilla tai vastaavilla. Tässä vaiheessa testataan myös järjestelmän eri ominaisuuksien toiminnollisuudet sekä erikseen että yhdessä. [Jenkins, 2008]

Järjestelmätestauksessa testataan koko järjestelmä ja sen käyttämät palvelut yms. kokonaisuutena [Jenkins, 2008].

Suorituskykytestausta voidaan tehdä eri testausvaiheissa. Alemmalla tasolla voidaan testata suoraan esimerkiksi jonkin algoritmin käyttäytymistä kuormituksen lisääntyessä. ylemmällä tasolla voidaan tutkia järjestelmän osien tai koko järjestelmän käyttäytymistä erilaisissa äärimmäisissä tilanteissa. Tämän avulla voidaan löytää mahdollisia heikkoja kohtia. [Jenkins, 2008].

Stabiilisuustestauksella tai stressitestauksella tarkoitetaan yleensä koko järjestelmän vakauden testaamista vaihtelevilla profiileilla ja kuormilla hieman pidempiä ajanjaksoja. Käytännössä yritetään simuloida järjestelmään kohdistuvaa kuormitusta ja käyttöä siten kuin se oikeassa käytössä tulisi olemaan. Tämän vaiheen avulla voidaan varmistua, että järjestelmä toimii luotettavasti myös pidempien testausjaksojen aikana [Patton, 2001].

Hyväksyntätestaus tehdään yleensä järjestelmän toimittajan ja asiakkaan toimesta ja jonka tarkoitus on todistaa, että järjestelmä toimii sille annettujen vaatimusten mukaisesti ja on valmis toimitukseen [Jenkins, 2008]. Asiakas itse yleensä määrittelee millaisista testeistä järjestelmän pitää kyetä suoriutumaan.

Eri vaiheiden testaussyklit vaihtelevat tarpeen, projektin vaiheen sekä käytettävän prosessin mukaan. Yksikkötestejä ajetaan mahdollisimman usein koodin aktiivisen kehityksen aikana, mutta myös säännöllisesti projektin kaikissa muissa vaiheissa.

Yksikkötestit ovat yleensä luonteeltaan lyhytkestoisia ja tällä saavutetaan se etu, että testien kesto ei ole esteenä sille, että niitä ajetaan toistuvasti. Yksikkötestit ovat usein myös melko helposti automatisoitavissa, joka omalta osaltaan varmistaa sen, että testejä on helppo toistaa aina tarvittaessa. Kuitenkin projektin edetessä yksikkötestien lukumäärän kasvaessa, voidaan päätyä tilanteeseen, jossa kaikkien testien ajaminen esimerkiksi jokaisen käännöksen yhteydessä ei ole enää mielekäästä niiden liian pitkän keston takia. Yksikkötestien tulisikin olla mahdollisuuksien mukaan olla ajettavissa yksittäin eli käytännössä olla riippumattomia toisista yksikkötesteistä. Näin ollen yksikkötestejä voidaan esimerkiksi ryhmitellä ja näin ollen ajoaika saadaan riittävän lyhyeksi.

Hajautetussa työympäristössä erityisesti integrointitestejä tulisi ajaa mahdollisimman useasti. Käytännössä testejä voidaan kuitenkin ajaa johonkin yhdessä sovittuun aikaan. Sopiva aika riippuu siitä, miten eri ryhmät tai henkilöt pystyvät toimittamaan tuotoksia. Integrointiajankohtaan vaikuttaa myös se, onko toimitetavalla sisällöllä riippuvuuksia muiden ryhmien tai henkilöiden tuotoksiin. Jos riippuvuuksia ei ole, niin tuotos voidaan integroida yhteisesti sovittuun aikaan. Otollinen aika tehdä integrointitestejä voisi olla esimerkiksi aina kun ryhmä tai henkilö lopettaa päivän urakkansa ja hänellä on jotain toimitettavaa järjestelmään. Näin ollen toiset saavat uusimmat versiot käyttöönsä päivän askareita aloittaessaan. Yleensä integrointitestausta suoritetaan siihen aikaan, kun se vähiten haittaa koodin integrointia ja kehitystä. Mitä hajautetumpaa ohjelmistokehitys on, sitä hankalampaa on löytää sopiva aika koodin integroinnille ja testaamisella. Integrointitestaamisen automatisointi helpottaa testien ajoa esimerkiksi öiseen aikaan tai muuten hiljaiseen aikaan. Integroinnin avuksi löytyy myös hyviä työkaluja, kuten Gerrit [Gerrit, 2015], joka on katselmointia varten tarkoitettu työkalu, mutta johon voidaan liittää erilaisia ”koukkuja”, joiden avulla voidaan käynnistää esimerkiksi yksikkötestejä, koodin staattista analysointia tai mitä ikinä voidaan automatisoida melko yksinkertaisesti.

Järjestelmätestien ajaminen painottuu usein projektin loppupuolelle, kun ohjelmisto on saavuttanut riittävän kypsyyssasteen. Myös stabiilisuustestejä ajetaan pääasiallisesti

projektin loppupuolella. Suorituskykytestejä olisi hyvä ajaa jo mahdollisimman varhaisessa vaiheessa, heti sen ollessa mahdollista, jotta mahdollisiin ongelmakohtiin voitaisiin puuttua jo varhaisessa vaiheessa. Myöhemmässä vaiheessa havaitut suorituskykyongelmat saattavat kumuloida paljonkin ylimääräistä työtä, joka tarpeettomasti siirtää tuotteen valmistumista.

Testaustulosten jakaminen hajautetussa työympäristössä on ensiarvoisen tärkeää. Erityisesti integrointitestauksen tulokset tulisi saattaa kaikkien osapuolten saataville mahdollisimman pian, että mahdolliset ongelmat saadaan korjattua välittömästi. Integroinnissa ilmenevien ongelmien vuoksi viestinnän tarve kasvaa merkittävästi. Testaustulosten pitäisi olla riittävän kattavat, että niiden perusteella jokaisen ryhmänjäsenen pitäisi kyetä tekemään päätelmiä vian laadusta.

Asiakasvikojen syiden analysointiin käytetään analyysia: "Root Cause Analysis" (RCA) [Jenkins, 2008] ja Escape Defect Analysis (EDA). RCA:n tarkoitus on selvittää vian perimmäinen syy. EDA:lla varmistetaan, että ohjelmistokehitys- ja testausprosessit kehittyvät havaittujen puutteiden osalta [Vandermark, 2003].

### 3. Tapaustutkimukset

Tässä luvussa tarkastellaan kahta hieman toisistaan poikkeavaa tapaustutkimusta, joita yhdistää viestintä. Lisäksi käydään läpi muutamalla sanalla käytetty tutkimusmenetelmä ja tutkimusaineisto sekä hieman kahden tapaustutkimuksen välistä vertailua.

#### 3.1 Tutkimusmenetelmä ja tutkimusaineisto

Tässä tutkimuksessa on käytetty kvalitatiivista tutkimusmenetelmää. Tutkimusstrategiana on käytetty case-tutkimusta eli tapaustutkimusta ja tutkimusasetelma on ex-post facto eli tapahtuman jälkeinen tarkastelu. Tapauksia peilataan myös omiin aikaisempiin kokemuksiin sekä kirjallisuuteen.

Tutkimusaineisto koostuu pääasiallisesti omiin kokemuksiin sekä havaintoihin kyseisissä tapauksista. Tämä johtuu luonnollisesti tutkimusasetelmasta, jossa tutkitaan jo tapahtuneita asioita ja pohditaan miten asiat omasta näkökulmasta tapahtuivat. Tapaus yhden kohdalla omat kokemukset eivät rajoitu ainoastaan noihin muutamaan viikkoon, vaan useamman vuoden työskentelyyn kyseisessä ympäristössä. Tapauksessa kaksi kokemusta ympäristöstä ja tavasta tehdä töitä kertyi melkein kahden vuoden ajalta. Mitään erillistä rakenteellista tai rakenteettomia haastatteluja ei järjestetty, mutta erinäisiä keskusteluja tapauksiin liittyen on käyty. Keskustelut ovat olleet vapaamuotoisia ja käyty ilman muodollista rakennetta muistuttaen lähinnä kahvipöytäkeskusteluja. Yhteistä keskusteluille on myös se, että ne on käyty tapausten ulkopuolella, jonkin aikaa tapausten jälkeen.

Tapaukset tähän tutkimukseen on valittu tiettyjen ominaisuuksien takia. Molemmat tapaustutkimukset liittyvät läheisesti tiedon ja tietämyksen siirtämiseen, yleisesti voidaan tapausten sanoa liittyvän tavalla tai toisella viestintään organisaation sisällä, että organisaation ulkopuolelle sekä siihen liittyviin ongelmiin. Molempiin tapauksiin liittyy myös maantieteellisen hajaantuneisuuden tuomat lisähaasteet. Eräs mielenkiintoinen seikka liittyen viestintään oli se, että ensimmäiseen tapaustutkimukseen liittyvät henkilöt olivat vuosien takaa tuttuja, kun taas toisessa tapaustutkimuksessa henkilöt olivat ennalta tuntemattomia. Ensimmäinen tapaustutkimus sijoittuu ajallisesti 2011 vuodelle ja toinen tapaustutkimus vuodelle 2013.

Tapaustutkimuksia analysoitaessa painotetaan kirjallisuudesta valittuja elementtejä ja peilataan niitä omiin havaintoihin valitun aikavälin puitteissa. Sopiviksi elementeiksi valitaan seuraavat:

1. Viestintä
2. Kulttuuri ja kielimuuri
3. Tietämyksenhallinta ja tiedon siirto
4. Ryhmän suorituskyky
5. Virtuaalitiimi (vain tapaus 1)
6. Ohjelmiston laatu

Ensimmäinen tapaustutkimus keskittyy lähinnä tietämyksen siirtämiseen (competence transfer) tilanteessa, jossa tehtävät siirretään organisaation sisällä halvempien kustannusten toivossa. Tapaustutkimus keskittyy kolmen viikon ajan jaksoon ja kasvokkain tapahtuvalla viestinnällä oli tässä merkittävä rooli. Toinen huomion arvoinen seikka on se, että koska tapauksessa kysymys oli ulkoistamiseen liittyvistä tapahtumista, siihen liittyi vahvasti tunteet sekä muutosvastarintaa.

Toisessa tapauksessa on myös kyseessä tiedon siirtämisestä. Erona ensimmäiseen tapaukseen verrattuna olivat muun muassa se, että tässä tapauksessa itse havainnoitsijana olin vastaanottamassa tietoa ja tietämystä. Kun toisessa tapauksessa itse havainnoitsijana olin siirtämässä tietoa. Myös viestintään käytetty teknologia erosi ensimmäisestä tapauksesta siten, että suurin osa viestinnästä pohjautui muuhun kuin kasvokkain käytävään viestintään. Tässä tapauksessa tehtävää varten oli perustettu virtuaalitiimi. Virtuaalitiimi ja sen toiminta ovatkin vahvasti mukana tässä tapaustutkimuksessa.

Molemmille tapauksille on yhteistä se, että itse havainnoitsijana työskentelin molemmissa tapauksissa alihankkijan roolissa. Tämä vaikuttaa viestinnän näkökulmasta esimerkiksi siihen, miten eri organisaatioissa rajataan organisaation sisäinen ja ulkoinen viestintä sekä mihin tietoihin alihankkijalla on pääsy.

Kumpaakin tapaustutkimusta tarkastellaan tiettyjen ominaisuuksien suhteen, jotka pääasiallisesti liittyvät viestintään ja pohjautuvat kirjallisuudessa tehtyihin havaintoihin, omiin havaintoihin kyseisistä tapauksissa sekä työhistorian kautta saatuihin kokemuksiin.

### 3.2 Tapaus 1: työn siirtäminen kustannussyistä

Ensimmäinen tapaus liittyy erääseen kansainvälisillä markkinoilla toimivaan suomalaislähtöiseen ohjelmistopalveluita alihankintana tarjoavaan yritykseen, kutsuttakoon tätä yritystä vaikkapa nimellä Yritys Y. Yrityksellä on toimintaa eripuolella



maailmaa, myös Suomessa. Tapauksessa Y toimi alihankkijana eräälle mobiileita verkkoteknologiaratkaisuja tarjoavalle yritykselle, kutsuttakoon tätä yritystä vaikkapa Z:ksi. Y on toiminut läheisessä yhteistyössä jo usean vuoden ajan Z:n kanssa. Kilpailun yhä kiristytessä kansainvälisesti mobiilien verkkoratkaisuiden rintamalla, Y tarjosi Z:lle ratkaisua, jossa projektissa työskentelevä Y:n suomalainen työvoima korvattaisiin asteittain Y:n oman organisaation Kiinassa toimivalla työvoimalla. Näin ollen kustannuksiin saataisiin säästöjä suoraan halvempien työvoimakustannusten kautta. Matkapuhelinverkkoteknologia on siinä mielessä erittäin säädeltyä, että standardit määrittelevät melko tarkkaan, miten verkkoelementtien tulee toimia. Yleensä yksi verkkolaitetoimittaja ei toimita järjestelmän kaikkia osia, vaan ne tulevat kahdelta tai useammalta laitetoimittajalta. Y:n suomalaisilla työntekijöillä oli keskimäärin reilusti yli kymmenen vuoden työhistoria ohjelmistoalalta ja vankka kokemus matkapuhelinverkkojen ohjelmistosuunnittelusta sekä testauksen eri osa-alueista. Mainittakoon myös, että suurin osa suomalaisista oli työskennellyt yhdessä samoissa projekteissa vuosien saatossa ja näin ollen he tunsivat toisensa erittäin hyvin. Sen sijaan Y:n kiinalaisilla työntekijöillä ei ollut mainittavaa kokemusta matkapuhelinverkoista eikä merkittävää työhistoriaa ohjelmistoalalta vaan osa oli vastavalmistuneita ja tämän oletuksen pohjalta heillä ei ollut merkittävää yhteistä kokemuspohjaa.

Itse työskentelin alihankkijana Y:ssä ohjelmistosuunnittelijan roolissa, tosin tehtäviini kuului myös testauspuolen tehtäviä sekä pääsin seuraamaan läheltä käytännön toteutusta tietämyksen siirtämisestä osapuolelta toiselle. Töiden ja tietämyksen siirtäminen tapahtui useassa eri vaiheessa, aina valmisteluvaiheista käytännön tietämyksen siirtämisen kautta itse töiden siirtoon asti. Tässä kohdassa kuvataan ainoastaan tapahtumat, jonka aikana tietämyksen siirtäminen oli vaiheessa, jossa kiinalaiset kollegat olivat kolmen viikon ajan vierailulla Suomen toimipisteellä. Tämän kohdan tarkoituksena on kertoa esimerkkejä käytännöistä, miten tietoa ja tietämystä pyrittiin jakamaan organisaation sisällä.

Y:n suomen organisaatiosta alihankintaa Z:lle teki parhaimmillaan kuusi scrumtiimiä. Y:n kiinalaiset kollegat vierailivat Suomessa kolmena eri ajankohtana, kolmen viikon ajan per kierros. Jokainen ryhmä koostui kuudesta henkilöstä, jotka edustivat kahden eri scrumtiimin jäseniä. Kustakin tiimistä mukana olivat scrummaster, arkkitehti/ohjelmistosuunnittelija sekä ohjelmistotestaja. Ryhmänjäsenien tehtäviin kuului myös toimia eräänlaisina sillanrakentajina tai viestinvälittäjinä. Heidän tehtävänä oli hoitaa päivittäin viestintä Kiinaan oman scrumtiimin suuntaan sekä luoda kontaktiverkostoa suomen suuntaan. Etukäteen sekä suomalaisille että kiinalaisille tarjottiin lyhyt oppimäärä toisen maan kulttuuriin. Etukäteislukemiseksi kiinalaiset kollegat olivat saaneet valikoituja dokumentteja muun muassa järjestelmän arkkitehtuurista sekä ominaisuuskuvauksia. Vierailun aluksi Suomen scrumtiimien

arkkitehdit kävivät vieraiden kanssa vielä läpi saman materiaalin, mihin he olivat tutustuneet jo etukäteen. Näin yritettiin varmistautua, että mahdolliset ja myös todennäköiset väärinymmärrykset voitiin mahdollisimman aikaisessa vaiheessa oikaista.

Ensimmäisessä yhteisessä virallisessa tilaisuudessa kiinalaiset ja suomalaiset kollegat esiteltiin toisilleen sekä scrumtiimit, jossa heidän oli tarkoitus tehdä projektitöitä muun tiimin avustuksella. Näin oli tarkoitus työn kautta oppia tuntemaan lähemmin kehitys- ja testausympäristöä. Viikon pikaopintojen jälkeen Kiinan scrumtiimien oli tarkoitus osallistua virtuaalisesti yhden sprintin töihin, jonka sisällöksi valittiin sama, kuin jonkun Suomessa toimivan tiimin oli tarkoitus toteuttaa. Tämän katsottiin olevan hyvä tapa perehdyttää kiinalaiset sekä tapaan tehdä töitä, että tutustuttaa heitä pintapuolisesti johonkin valittuun ominaisuuteen. Sitä myöden koodi sekä sen testaaminen tulivat myös tutuksi. Learn by doing – lähestymistavan katsottiin olevan parempi tapa oppia asioita ja pitää mielenkiinto yllä kuin pelkkään koodiin tutustuminen lukemalla. Näin myös siksi, että lähdekoodia oli kertynyt parissa vuodessa jo melko paljon. Myöskään toimintalogiikan seuraaminen tilakoneessa ei ole aina kovin suoraviivaista, ainakaan jos ei tunne sovellusaluetta perinpohjaisesti. Tämän tyyppinen lähestymistapa mahdollisti myös tavan tarjota helpommin vertaistukea, koska molemmat tiimit tekivät samoja tehtäviä. Luonnollisesti muutaman viikon intensiivisetkään opinnot eivät voi tarjota muuta kuin hyvän alustan oppia jatkossa lisää. Haasteet oppimisessa liittyvät muun muassa kehitys- ja testausympäristön omintakeisiin piirteisiin sekä järjestelmän kompleksisuuteen. Sovellusalueen syvällisempi tunteminen vie helposti kuukausia, vaikka matkapuhelinverkkoteknologia olisikin ennestään tuttua, pidempään jos teknologia ei ole tuttu.

Yhden sprintin harjoitus oli kahden viikon mittainen ja sen sisältö oli karkeasti seuraava: Ensimmäisenä päivänä scrumtiimeille jaettiin omat storyt. Story on käytännössä jokin kokonaisuus, joka on etukäteen arvioitu olevan sisällöltään sopiva seuraavan kahden viikon ajanjaksolle. Scrumtiimin jäsenet itse arvioivat työn määrän ja jakavat ne pienempiin osiin. Optimaalisin koko yhdelle tehtävälle katsottiin olevan yhden päivän työ. Jos tiimi katsoi, että työ oli mahdollista saada kokonaisuudessaan valmiiksi sprintin aikana, sitoutui tiimi tehtävään. Tiimin vastuulla oli myös työn integrointi järjestelmään, testata toteutettu ominaisuus toimivaksi sekä esitellä ominaisuus koko projektille sprintin päätteeksi. Jos tiimi taas katsoi, ettei työtä voi saada valmiiksi, oli sen tehtävänä pilkkoa tehtävät sopiviin osiin ja muodostaa niistä uudet storyt, jotka joko hyväksyttiin tai hylättiin aikanaan. Pienen kokonaisuuden pilkkominen helposti hahmotettaviin osiin tarjosi kiinalaisille mahdollisuuden oppia tavasta työskennellä lyhyissä sykleissä. Harjoitus pakotti hahmottamaan myös kokonaisuutta sekä mihin toteutettava osa sijoittuu kompleksisessa järjestelmässä.

On täysin mahdotonta edes arvella, miten tiedon välitys Suomessa olevien kiinalaisten ja Kiinassa olevien kollegoiden välillä sujui. Paras arvaus on, että ongelmia oli. Arvaus perustuu ainoastaan siihen tietoon, mitä myöhemmin Kiinassa käyneet kollegat ovat asiasta kertoneet sekä siihen, että laadun nostaminen vaaditulle tasolle vei oletettua pidempään.

Lopuksi yhteenvetona laadullinen arviointi, miten töiden ja tietämyksen siirtäminen loppujen lopuksi sujui. Laadullisen arvioinnin pohjaksi valitaan tähän tutkimukseen kirjallisuudesta löydetty teoriat ja tutkimukset, jotka parhaiten sopivat tämän tyyppisen tiedon siirtämisen arviointiin. Arvioinnissa käytettiin seuraavia ominaisuuksia: Kulttuuri erot, kielimuuri, tiedon siirtäminen, fasilitetit, ryhmänsuorituskyky sekä ryhmien välinen yhteisymmärrys.

Ensimmäiseksi arvioidaan kulttuurisia eroja (kansalliset, organisaatio, ammatillinen). Kirjallisuudessa on esitetty, että projektin alulla on suuri merkitys ryhmän jäsenten välisen tuttuuden luomiseksi. Projektin alussa suositellaan pitämään esimerkiksi koko projektin ”kickoff”-tilaisuus tai vastaava. Sen tarkoituksena on ensisijaisesti jakaa projektiin liittyvää tietoa kaikille, mutta myös tutustuttaa ryhmän jäsenet toisiinsa edes pintapuolisesti. Se, että pelkkä nimi saa kasvot helpottaa luottamuksen luomisessa sekä epäluulojen hälventämisessä. Tämän case studyn osalta projektin ”kickoff”-tilaisuutta ei pidetty ainakaan siinä muodossa, että paikalla olisivat olleet molemmat osapuolet. Etukäteen Suomessa järjestettiin lyhyt vieraisiin kulttuureihin perehdyttämiskurssi. Kurssin sisältö koostui lähinnä pintapuolisista esimerkeistä mahdollisista ongelmista mitä kulttuurierot saattavat aiheuttaa. Selvimmät kulttuurieroista johtuvat ongelmat alussa olivat kielimuuri, joka oli merkittävin haaste, luottamuspula, muutosvastarinta sekä asenne (tulevat ja vievät meidän työt). Kirjallisuus tunnistaa edellä mainitut ongelmat. Kolmen viikon lähioppimisjakson lopussa järjestettiin yhteinen retki luonnon helmaan. Retken pääasiallinen anti oli lähinnä tutustua toisiin paremmin ruuan laittamisen, erilaisten pelien, soutamisen ja saunomisen ohessa. Tosin saunominen ja uiminen osoittautuivat vieraille ylitsepääsemättömäksi esteeksi, mutta soutaminen ja viinan juonti tuntuivat olevan mieluisampia harrasteita. Ruokaa valmistettiin siten, että kiinalaiset valmistivat omat bravuurinsa ja suomalaiset omansa. Vaikka tämän tyyppiset tapahtumat tuntuvatkin välillä olevan aivan turhaa rahan haaskausta, niin todellisuudessa ne ovat enemmän kuin oiva tilaisuus tutustuttaa ihmisiä toisiinsa. Tilanne ja paikka ovat yleensä kaikille outoja ja jo se hieman tasoittaa henkilöiden välillä olevia esteitä. Koska ohjelman sisältö ei yleensä ole sidottu tiukasti työhön, niin myös se edesauttaa rennomman ilmapiirin syntymiseen. Tilanne voi tosin kääntyä itseään vastaan, jos joukossa ei ole yhtään henkilöä joka luonnostaan tai viran puolesta toimii jäänmurtajana. Tässä

tapauksessa joukostamme löytyi kirjallisuuden kuvaama ”sillanrakentaja”-roolin omaava henkilö. Kirjallisuus puoltaa myös sitä, että neutraalilla maaperällä on helpompi tehdä tuttavuutta sekä keskustella muistakin kuin työasioista. Nämä saattaa omalta osaltaan auttaa hälventämään ennakkoluuloja ja negatiivisia asenteita. Huolimatta siitä, että edellä kuvatun kaltainen retki on erinomainen tapa lieventää kulttuurieroista aiheutuvia ongelmia, niin ajankohta ei ollut oikea. Kuten kirjallisuus suosittelee, kannattaa projektin alkuun panostaa juurikin tuttuuden luomiseen sekä muiden viestintää helpottavien toimien kautta.

Toiseksi arvioidaan kielimuurin vaikutusta. Kielen vaikutus viestintään, kuten kirjallisuus toteaa, oli tässä tapauksessa ilmeinen. Yhteisen kielen löytäminen oli aluksi melko haastavaa. Tähän syynä ei ehkäpä ollut pelkästään heikko suullinen kielitaito, vaan kielen suullisen taidon käyttämättömyys. Suomessa työskentelevä ryhmä teki kylläkin töitä hajautetussa ympäristössä, mutta harvemmin järjestettiin yhteisiä palavereita, jossa olisi ollut tarvetta käyttää kieltä. Normaalisti käytössä olivat sähköposti ja pikaviestimet. Aika loivensi kielen aiheuttamia ongelmia, koska opimme kuuntelemaan toinen toisiamme ja muutaman viikon kuluttua ymmärsimme toisiamme jo melko hyvin. Yhteisen termistön ja sanaston puute oli myös aluksi ongelmallista, koska se vaikeutti tietämyksen siirtämistä ryhmältä toiselle.

Kolmanneksi arvioidaan tiedon siirtämistä sekä täsmällisen tiedon, että hiljaisen tiedon siirtämistä. Täsmällinen tieto voidaan kirjallisuuden mukaan tallentaa ja jakaa edelleen. Tässä tapauksessa tarjolla oli jo etukäteen luettavaksi joitain dokumentteja, kuten yleisen tason järjestelmä arkkitehtuurikuvauksia ja kuvaksia järjestelmän ominaisuuksista. Täsmälliseksi tiedoksi voidaan myös lukea käytettyjen tekniikoiden, kuten ohjelmointikielen oppiminen. Hiljaisen tiedon kirjallisuus taas määrittelee tiedoksi, jota harvoin, jos ollenkaan voidaan saattaa kirjalliseen muotoon ja jakaa eteenpäin. Tässä tapauksessa erityisesti, koska järjestelmä pohjautui osittain hyvinkin vanhaan perinnejärjestelmään. Tämän kaltaisen osittain vanhoihin perintöihin pohjautuvien järjestelmien tietämys on usein, niin myös tässä tapauksessa, hiljaisena tietona yksilöiden hallussa. Dokumentoitua tietoa toki löytyy paljon, mutta tietoa miten asiat kannattaa tehdä, ei kirjoista ja kansista löydy. Tämän tiedon siirtäminen osoittautui melko haasteelliseksi. Esimerkkinä mainittakoon, että tämän kokoisen järjestelmän testaaminen jo integrointitasolla vaatii laajaa järjestelmätuntemusta, jota vierailijoilla ei ollut.

Neljänneksi arvioidaan työskentelytilan vaikutusta tiedonsiirtämiseen, toimiko avotila niin kuin edeltävissä tutkimuksissa on todettu? Kirjallisuus useasti ehdottaa, että avoin toimistotila parantaa ryhmän sisäistä viestintää. Tämä voitiin todeta tässäkin projektissa, tosin sillä erotuksella, että ryhmästä muodostui ensin kaksi ryhmää, suomalaiset ja

kiinalaiset ja näiden ryhmien sisäinen viestintä toimi ainakin päällisin puolin moitteetta. Kirjallisuuden ehdottamat spontaanit käytäväkeskustelut ja rönsyilevä kahvipöytäkulttuuri loistivat poissaolollaan. Tässäkin tapauksessa spontaaneita keskusteluja käytiin ja kahvia juotiin, mutta näissäkin keskusteluissa voitiin havaita kulttuurien välinen kuilu. Uudemmat tutkimukset tosin väittävät, että avokonttorin haitat ovat niiden hyötyä suuremmat, kuten melun aiheuttamat ongelmat sekä yksityisyyden puute. Osittain nämä väitteet on helppo myöntää olevan totta, mutta ainakin tässä tapauksessa melu ei ainakaan itseäni haitannut, yksilöllistä tosin tämäkin.

Viidenneksi arvioidaan ryhmänsuorituskykyä, vertaillaan kokeneemman ja kokemattomamman ryhmän eroja ja onko kirjallisuudessa havaitut väitteet tämän yksittäisen tapauksen kannalta merkittäviä. Kirjallisuus ehdottaa, että ryhmän jäsenten välinen tuttuus edesauttaa ryhmän suorituskykyä, jos ryhmän koordinointi on vaikeaa ja tehtävien tuttuuden vaikutus ryhmän suorituskykyyn laskee, jos tehtävät ovat haasteellisempia. Tässä tapauksessa suomessa toimivat ryhmät olivat toisilleen tuttuja jo useampien vuosien takaa eri projektien yhteydestä ja kiinalaiset olivat tehneet töitä keskenään vasta lyhyen ajan. On erittäin vaikeata sanoa miten ryhmän jäsenten välinen tuttuus vaikutti ryhmän suorituskykyyn, jos asiaa tarkastellaan ainoastaan ryhmän koordinoinnin haasteellisuuden pohjalta. Voin ainoastaan arvailla, että tuttuus olisi voinut auttaa suorituskyvyn paranemisessa, jos osa ryhmästä olisi siirtynyt tekemään töitä toisaalle. Kiinalaisille osan ryhmästä oleminen etäällä (kiinassa) voi antaa viitteitä siitä, että vasta vähän aikaa yhdessä toimineen ryhmän suorituskykyyn ryhmän tuttuudella saattaa olla vaikutusta. Tämäkin voidaan todeta epäsuorista havainnoista. Jälkeenpäin voitiin todeta, että ilmeisesti kaikki tieto ei kovasta yrittämisestä huolimatta saavuttanut kuulijaansa, joka näkyi työn heikkona laatuna.

Kuudenneksi ja viimeiseksi arvioidaan ryhmien välisen yhteisymmärryksen tai sen puutteen mahdollisia vaikutuksia tiedon siirtämiseen ja verrataan kirjallisuudessa tehtyihin havaintoihin. Kirjallisuuden mukaan hyvällä ryhmähengellä tai yhteisymmärryksellä on positiivinen vaikutus tiedon siirtämiseen. Tämän projektin lopputuleman perusteella tässä tapauksessa vaatimattomammalla ryhmähengellä ja yhteisymmärryksellä saattoi olla tiedon siirtoa heikentävä vaikutus. Tältä osin havainto on yhtenevä kirjallisuuden olettamukseen.

Vaikka tästä kolmen viikon aikaikkunasta ei juuri voi tehdä suuria johtopäätöksiä miten tietämyksen siirtämisessä olleet haasteet vaikuttivat ohjelmiston tulevaan laatuun. Jälkeenpäin olen kuullut, että laatu ei ollut aivan kohdallaan. Suurimmat ongelmat aiheuttivat monimutkaisen järjestelmän testaaminen sekä osaltaan käytetyt työkalut ja

prosessit. Testien määrää lisäämällä ei välttämättä kompensoida puutetta tiedon puutetta domain-osaamisessa.

### 3.3 Tapaus 2: yhtenäiset testauskäytännöt virtuaalitiimin avulla

Tässä kohdassa kuvataan erään organisaation virtuaalitiimin toimintaa. Virtuaalitiimi oli perustettu, jotta testausaktiviteetit saataisiin yhtenäistettyä erään tuotteen parissa työskentelevien tiimien kesken. Tämän kappaleen tarkoituksena on kuvata tiedon ja tietämyksen jakamista organisaation sisällä tilanteessa, jossa osapuolet ovat hajaantuneet maantieteellisesti eri paikkoihin.

Tämän esimerkkitapauksen yritys on kansainvälinen suuryritys, jolla on toimintaa ympäri maailman, myös Suomessa. Kyseisessä Suomen toimipisteessä yritys on toiminut vasta muutaman vuoden ajan. Yritys toimii sulautettujen järjestelmien parissa, niin raudan kuin ohjelmiston toimittajana, kutsuttakoon tätä yritystä Yritys X:ksi. X:llä on omia tuotteita ja erittäin laajasti tuotekehitystä sekä korkea teknologinen osaamista. X:llä on myös runsaasti kokemusta virtuaalitiimien hyödyntämisessä organisaatiossaan.

Tiimi perustetaan aina jotain tiettyä tehtävää varten, näin on myös virtuaalitiimin kanssa. Virtuaalitiimi perustetaan tehtävää varten ja tehtävän suoritettuaan virtuaalitiimin toiminta päättyy. Tässä tapauksessa virtuaalitiimi perustettiin siksi, että saman tuotteen eri tiimit ottaisivat käyttöönsä samoja testaukseen liittyviä työkaluja ja testauskäytäntöjä. Jokaisen tiimin tulisi myös omalta osaltaan toteuttaa kaikki vaaditut testivaiheet sekä tarjota testitulokset määrämuodossa niiden esittämistä varten. Virtuaalitiimin kautta pyrittiin myös jakamaan muille tiimeille tiimeissä hyviksi havaittuja testauskäytäntöjä. Lähtökohtana oli, että saman tuotteen eri osa-alueiden testaajat eivät juurikaan tehneet yhteistyötä ja näin ollen käytännöt testata tuotteen eri osa-alueita olivat melko kirjavat. Myös testausympäristöjen suhteen tiimien välillä oli huomattavia eroja. X:n sisällä työskenteli myös laaja joukko alihankkijoita, joka omalta osaltaan oli vaikeuttamassa tietojen ja käytäntöjen vapaata leviämistä tiimien välillä. Eräänä sivujuonteena virtuaalitiimin jäsenille tarjottiin X:n kehittämää toteutusta "Model based testing" –konseptista. MBT-alusta on avoimeen lähdekoodiin perustuva ja sen perustavaa laatua oleva idea on generoida ja ajaa testit täysin automaattisesti sekä luoda myös sellaisia testitapauksia, "polkuja", joita testaajat eivät normaalisti testaisi. Menetelmän tarkoituksena on säästää aikaa ja rahaa sekä lisätä testien kattavuutta.

Yleisesti virtuaalitiimin voisi kuvailla olevan lyhytaikainen ja tiettyä tehtävää varten koostettu joukko asiantuntijoita, jotka ratkaisevat annetun ongelman ja siirtyvät kukin eteenpäin seuraaviin haasteisiin. Esimerkkitapauksessa virtuaalitiimin perustaminen otti

kuitenkin aikansa ja sen toiminta ei ollut aluksi kovinkaan aktiivista. Virtuaalitiimi koostui yhden tuotteen eri osa-alueiden testaajista, yksi per tiimi sekä tuotteen arkkitehteistä, virtuaalitiimin vetäjästä sekä pienestä joukosta asiantuntijoita. Virtuaalitiimi oli myös kasassa melko pitkään, johtuen ehkäpä siitä, että virtuaalitiimi ei ollut erityisen aktiivinen sekä virtuaalitiimin tehtävä ei ollut aivan kristallinkirkas heti alusta lähtien.

Ensimmäinen palaveri käytiin virtuaalitiimiin vetäjän sekä muutaman testauksen asiantuntijan kanssa. Palaveri käytiin konferenssipuhelimen välityksellä. Palaverin aiheet koskivat lähinnä testaamiseen liittyviä yleisiä käsitteitä ja niiden läpi käymistä. Näin varmistuttiin siitä, että jokaisella virtuaalitiimin jäsenellä olisi jokseenkin yhtenevä käsitys testaamisesta ja siihen liittyvästä terminologiasta. Virtuaalitiimin vetäjä keräsi myös tiimin sen hetkisen tilanteen, mitä testataan, miten testataan sekä jätetäänkö jotain tietoisesti testaamatta ja onko testaamatta jättämiseen jokin erityinen syy, kuten työkaluista, testausympäristöstä tms. johtuvat ongelmat tai puutteet. Virtuaalitiimin jäsenen tuli työskennellä X:n oman organisaation sisältä, eikä alihankkijoita hyväksytty tiimin jäseneksi. Itse alihankkijana jouduin heti hankalaan tilanteeseen, koska oman tiimini esimies oli sitä mieltä, että ketään muuta ei moiseen kerhoon tiimistä liikene kiireeseen vedoten ja virtuaalitiimin vetäjä taas oli sitä mieltä, että politiikka ei salli alihankkijoiden osallistumista virtuaalitiimin toimintaan. Lopputuloksena oli se, että osallistuin soveltuvin osin virtuaalitiimin toimintaan. Myöhemmin osoittautui kuitenkin, että pystyin osallistumaan täysipainoisesti virtuaalitiimin toimintaan, ainoastaan matkustaminen oli alihankkijoilta tiukasti kielletty.

Virtuaalitiimi kokoontui keskimäärin kerran viikossa konferenssipuhelimen välityksellä ja aluksi tavoite oli pitää palaveri lyhyenä noin puolen tunnin mittaisena. Palaverin agenda oli saatavilla etukäteen virtuaalitiimin kotisivuilta, jonne kerättiin muutenkin tarpeellista tietoa tavoitteisiin liittyen. Viikkopalaveri koostui lähinnä yleisten tavoitteiden läpikäynnistä sekä jokaisen tiiminjäsenen oman tiimin testauksen statuksen läpikäymisestä. Äänessä oli miltei yksinomaan virtuaalitiimin vetäjä. Palavereista päällimmäisenä jäi mieleen lähinnä terminologian epäjohdonmukainen käyttö sekä konferenssipuheluiden huonolaatuinen ääni sekä kielen aiheuttamat ongelmat. Satunnaisesti virtuaalitiimin vetäjä vieraili virtuaalitiimin jäsenten luona kyselemässä kuulumisia. Päättävöitteeksi virtuaalitiimille määrittyi testausvaiheiden yhtenäistäminen, testaustyökalujen harmonisointi, testauksen automatisointi sekä testaustulosten julkistaminen määrämittäisessä muodossa. Testaustulokset olivat kaikkien nähtävillä muun muassa toimiston kahvihuoneissa. Testaustulokset esitettiin ns. laavalamppu näkymän avulla, josta kävi nopeasti ilmi, minkä tuotteen osa-alueen testit saavuttivat hyväksyttävän tason. Laavalampun punainen väri ilmaisi sen, että liian suuri määrä testejä epäonnistui. Keltainen väri ilmaisi sitä, että testit tulisi suorittaa välittömästi

uudelleen. Vihreä väri ilmaisi sen, että ajetut testit saavuttivat hyväksyttävän tason. Testitulokset vanhenivat jonkin ajan jälkeen, jolloin väri muuttui ensin keltaiseksi ja lopuksi laavalamppu sai punaisen värin. Testien kattavuus oli myös yksi asia, josta jokaisen tiimin tuli kerätä yhteenveto. Kattavuuksille oli määritelty alarajat, joista sai poiketa vain asiallisin perustein. Tiimien käyttöön tarjottiin myös muita työkaluja, kuten koodin staattiseen analysointiin käytettyä työkalua.

Reilun puolen vuoden toiminnan jälkeen, virtuaalitiimin yhteisten testauspäivien suunnittelu vei suurimman osan virtuaalitiimin viikoittaisesta palaveriajasta. Testauspäivät järjestettiin Suomessa, joten paljon aikaa kulutettiin agendan hiomiseen ja käytännön järjestelyiden sopimiseen. Testauspäivien tarkoituksena oli tutustuttaa virtuaalitiimin jäsenet keskenään sekä luoda yhteistä pohjaa yhteistyön helpottamiseksi jatkoa ajatellen. Itse en tilaisuuteen osallistunut, mutta kollegani tiimistä osallistui ja häneltä kuulin tilaisuuden kulusta. Tilaisuus koostui lähinnä eri tuotteiden tiimien testaajien tai arkkitehtien pitämistä esityksistä sekä vapaamuotoisesta iltahjelmasta.

Laadullisen arvioinnin pohjaksi valitaan tämän tutkimukseen kirjallisuudesta valikoituneista teorioista ja tutkimuksista ne, joita voidaan parhaiten soveltaa arvioinnissa, jossa tarkastellaan yhtenäisten testauskäytäntöjen käyttöön ottaminen useamman tuotteen kesken. Arvioinnissa käytettiin seuraavia ominaisuuksia: Tiedon jakaminen, virtuaalitiimin toiminta, ohjelmiston laatu sekä käytettyjen työkalujen hajautetussa projektissa.

Ensimmäiseksi arvioidaan miten *tiedon jakaminen* onnistui eli miten testauskäytäntöjen levittäminen onnistui hajautetussa ryhmässä sekä *viestintään* liittyviä ongelmia hajautetussa ryhmässä ja erityisesti *kielimuurin* aiheuttamia ongelmia, *etäisyydestä* aiheutuvia ongelmia ja *aikaerosta* aiheutuvia ongelmia Yleisesti kirjallisuudessaakin tunnistetut ongelmat liittyen viestintään ja tietämyksen siirtämiseen tulivat myös melko selkeästi esille tässä tapauksessa. Ensimmäisenä haasteena viestintään liittyen oli teknologian tuomat ongelmat, kuten konferenssipuheluiden heikko äänenlaatu, joka vaikeutti entisestään puheen ymmärtämistä. Kielestä johtuvat ongelmat korostuivat teknisten ongelmien myötä. Virtuaalitiimin jäsenten kielitaito oli melko hyvä, mutta välillä keskustelun seuraaminen oli silti haastavaa. Tiimissä oli jäseniä mm. Kiinasta, Ranskasta ja Suomesta. Aikaeron aiheuttamaa ongelmaa en itse huomannut, koska ainoastaan Kiinan jäsenet olivat Euroopan ulkopuolelta ja palaverit järjestettiin aina säädylliseen aikaan Euroopan virtuaalitiiminjäseniä ajatellen. Tosin aikaero-ongelma oli varmasti olemassa, vaikka se ei omaan työhöni vaikuttanut. Kulttuurien väliset erot eivät myöskään näytelleet kovin suurta osaa. Tämä saattoi johtua ainakin siitä tosiasiasta, että virtuaalitiimin jäsenten välinen kommunikointi oli oman näkemykseni mukaan melko



vähäistä. Viestintä keskittyi viikoittain järjestettävään yhteiseen palaveriin, mutta myöskin siihen tosiasiaan, että virtuaalitiimin vetäjällä oli mielestäni korostunut rooli viestin välittäjänä ja toiminnan organisoijana. Virtuaalitiimin vetäjä voidaan rinnastaa kirjallisuudessaakin havaittuun ”sillanrakentaja”-käsitteeseen.

Toiseksi tarkastellaan virtuaalitiimin toimintaa verrattuna kirjallisuudessa esitettyihin arvioihin. Vaikka X:ssä on historian saatossa käytetty virtuaalitiimejä erittäin yleisesti yhtenä organisaation muotona, niin kyseinen Suomen toimipiste oli ainoastaan muutaman vuoden vanha ja se mielestäni näkyi perustamisvaiheessa melko selkeästi. Virtuaalitiimin perustaminen tuntui kestävän melko pitkään ja oli epäselvää, mitkä tulisivat olemaan virtuaalitiimin tehtävät. Virtuaalitiimin tehtävä osoittautui melko laajaksi ja se edistyi melko hitaasti, joka johtui varmasti siitä, että eri tiimien lähtökohdat testaamisen suhteen olivat kovin erilaiset. Yhteisen taustan puuttuminen näyttäytyi mielestäni siinä, että virtuaalitiimi todella tuntui virtuaaliselta. Ehkäpä juuri tämän takia minkäänlaista mehenkeä ei tuntunut olevan. Vaikka kaikki ponnistelimme saman tuotteen eri osa-alueiden kimpussa, niin oma kuvani muiden tekemisestä jäi melko epäselväksi ja näin ollen tuli tunne, ettei muiden tekemisellä ollut vaikutusta omaan työhöni ja vaikutusta koko tuotteen valmistumiseen. Tämän kaltainen harhakuva vaikutti mielestäni myös siihen, että luottamuksella virtuaalitiimin jäsenten välillä ei tuntunut olevan suurtakaan merkitystä, koska muiden jäsenten tekemiset eivät tuntuneet vaikuttavan omaan suoritukseen ja onnistumiseen. Tähän vaikutti mielestäni myös yleinen ”sense of urgency” – johtamistapa muualla organisaatiossa, joka johti siihen, ettei virtuaalitiimin toimintaa aina nähty hyödyllisenä juuri siinä hetkessä ja näin ollen ainakin oma aktiivisuuteni jäi harmittavan alhaiselle tasolle. Oma näkemykseni on se, että sama päti myös muihin virtuaalitiimin jäseniin. Mielestäni virtuaalitiimin yhteinen tapaaminen olisi pitänyt järjestää jo paljon aikaisemmassa vaiheessa, näin ollen yhteenkuuluvuuden tunnetta, yhteistä päämäärää ja luottamusta olisi voitu vahvistaa ja sitä myöten yhteistyö jäsenten välillä olisi voinut olla aktiivisempaa. Vaikka virtuaalitiimin vetäjällä oli selkeästi päämäärät kirkkaana mielessään, mielestäni hän joutui ponnistelemaan omin voimin liian paljon eikä sama visio näyttäytynyt tiimin muille jäsenille ehkä riittävän selkeästi. Myös muun organisaation tuki tai oikeastaan sen puute vaikutti myös omalta osaltaan negatiivisesti virtuaalitiimin toimintaan. Virtuaalitiimin tulisi olla luonnostaan organisaation rajat ylittävä, mutta tässä tapauksessa alihankkijana tunsin olevani toisen luokan kansalainen. Tämä ei ollut omiaan kasvattamaan luottamusta muuhun organisaatioon ja päinvastoin. Lopuksi täytyy kuitenkin mainita, että mielestäni virtuaalitiimille on tilausta juurikin tämän tyyppiseen tarpeeseen, jossa halutaan esimerkiksi yhtenäistää käytäntöjä organisaation sisällä. Hyvin koostettu virtuaalitiimi tarjoaa erinomaisen mahdollisuuden päästä käsiksi laajempaan tietämykseen ja osaamiseen, kuin se olisi muuten mahdollista.

Kolmanneksi tarkastellaan ohjelmiston laatuun liittyviä seikkoja. Ohjelmiston laatu on tärkein asia, jonka asiakas kokee käyttäessään tuotetta. Ohjelmiston laatuun vaikuttavat useat eri asiat, mutta ohjelmistokehitysryhmän maantieteellinen hajaantuneisuus ei helpota asiaa. Kirjallisuuden mukaan hajautettu ohjelmistotyö perii kaikki keskitetyn ohjelmistoprojektin vaikeudet, minkä lisäksi vaikeudet ainoastaan korostuvat. Tärkeä tavoite yhteisten testauskäytäntöjen käyttöönotolla oli luonnollisesti ohjelmiston laadun parantaminen. Toinen vähintäänkin yhtä tärkeä seikka oli testauksen automatisointi sekä testitulosten jakaminen eri ryhmien välillä. Eräs kirjallisuudessa mainittu asia laadun parantamiseen ja tietämyksen jakamiseen on katselmoinnit. Hajautetussa ohjelmistosuunnittelussa katselmoinnit voidaan järjestää työkalujen avulla. Integrointi mainitaan myös eräänä ongelmakohtana hajautetussa projektissa. Tämä todettiin myös tämän projektin osalta todeksi. Puutteellinen viestintä aiheutti harmaita hiuksia, kun tuotoksia ryhmän eri jäseniltä integroitiin. Erityisen haasteellisia olivat toiselta mantereelta tulleet muutokset (patch). Kirjallisuus suosittelee agile-tyyppisessä kehityksessä, että mahdollisimman useasti integroidaan pieniä kokonaisuuksia. Näin ei ollut tässä tapauksessa, liekö syynä olleet epäselvät prosessit vai höltynyt kuri. Pahimmillaan yhdessä muutoksessa oli muutama tuhat riviä koodia ja lukematon määrä konflikteja.

Neljänneksi tarkastellaan kirjallisuudessa esitettyjä väitteitä ohjelmistojen tuesta hajautetun ryhmän välisessä viestinnässä sekä viestintävälineiden ongelmista. Kirjallisuudessa todetaan, että nykyään yhä useammat kehitystyökalut tukevat monimuotoisesti viestintää sekä läsnäolotietoa. Tämän projektin osalta kehitystyökalut olivat pääasiallisesti tekstipohjaisia työkaluja. Käytössä viestintään olivat lähinnä sähköposti, pikaviestimet sekä yleisimpänä konferenssipuhelut. Kirjallisuus toteaa sähköpostin lisäävän viivettä ja väärinymmärryksiä. Se voidaan varauksetta todeta myös tämän projektin osalta. Konferenssipuheluissa ja normaaleissa puheluissa ei viivettä ole, mutta kirjallisuus mainitsee ainakin huonon äänenlaadun lisäävän väärinymmärryksiä sekä turhautumista. Tämä voidaan ainakin osittain todeta olevan näin. Kirjallisuus toteaa myös aikaeron vaikuttavan negatiivisesti viestintään yhteisen ajan vähentyessä. Myös tämä voidaan todeta olevan käytännössä totta, vaikka virtuaalitiimin kohdalla aikaero ei vaikuttanut negatiivisesti. Muun ryhmän toimintaan aikaero vaikutti paljon enemmän, lähinnä yöllisinä tai aamuvaihain käytyjen puhelinneuvotteluiden osalta. Testien automatisointi sekä katselmointityökalut mahdollistivat muutosten ja testien tulosten tehokkaan jakamisen ryhmän kesken eri maantieteellisissä sijainneissa.

### 3.4 Tapaustutkimusten keskinäinen vertailu

Tässä kohdassa verrataan mitä mahdollisia havaintoja kahden tapaustutkimuksen välillä voidaan tehdä liittyen viestinnän onnistumiseen tai sen epäonnistumiseen sekä löytyikö tapausten välisistä havainnoista yhtäläisyyksiä tai eroavaisuuksia.

Molemmissa tapauksissa oli pohjimmiltaan kyse tiedon ja tietämyksen siirtämisestä organisaation sisällä, kahden tai useamman maantieteellisesti sijainnin välillä, jonka yhtenä lisämausteena tai haasteena viestinnässä oli alihankkija-elementti. Tiedon siirtäminen kahden tapauksen välillä erosi hieman, koska toisessa osallistuin tiedon siirtäjänä ja toisessa taas tiedon vastaanottajana. Myös tutkimuksen kannalta oleellinen huomio on, että toisessa tutkimuksessa viestintä perustui lähinnä intensiiviseen kasvokkain käytävään viestintään, kun taas toisessa tapauksessa viestintä oli verkkaisempaa ja se käytiin pääosin sähköisten viestimien avulla. Molemmissa tapauksissa tietämys saatiin siirrettyä vaikka toisessa projektissa (tapaus 2) ei varsinaista aikataulua ollutkaan. Tapauksessa kaksi suurin osa käytännöistä tuli käyttöön, mutta jälkepäin olen kuullut, että koko virtuaalitiimi kuihtui ajan myötä, kun sitä vetäneelle henkilölle tuli muita kiireellisempiä töitä. Tapauksessa yksi taas tietämyksen siirtämisessä pysyttiin, jos aikataulussa pysyminen katsotaan tavoitteen saavuttamiseksi. Loppujen lopuksi tuotosten lopullinen laatu kertoo enemmän kuin tuhat sanaa, miten molemmissa tapauksissa tietämys siirtyi ryhmältä toiselle. Tärkeä on myös huomioda se, että vaikka kasvokkain käytävän viestinnän katsotaan tarjoavan paremmat mahdollisuudet poistaa viestinnän esteitä, tosiasia on se, että ei sekään yksistään tee viestinnästä parempaa.

Kulttuurin vaikutus viestintään oli molemmissa tapauksissa ilmeinen. Kansallisen kulttuurin osalta suurempia haasteita voidaan havaita ensimmäisessä tapaustutkimuksessa. Tämä johtunee siitä seikasta, että tapauksessa yksi kulttuurien erot (Suomi-Kiina) tulivat selkeämmin esiin, kun taas tapauksessa kaksi virtuaalitiimi koostui lähinnä eurooppalaisista. Organisaatiokulttuurin osalta tapaus kaksi oli selkeästi haastavampi, koska organisaatiossa haluttiin erotella heidän omaan organisaation kuuluvat henkilöt ja organisaatioon kuulumattomat henkilöt, kuten alihankkijat. Tällä oli suoria vaikutuksia muun muassa siihen, että alihankkijalla ei ollut pääsyä tarvittaviin palveluihin, dokumentteihin, eikä alihankkijan osallistuminen esimerkiksi virtuaalitiimin toimintaan ollut itsestään selvyyttä. Epäsuorasti viestintään vaikuttavana asiana voidaan pitää sitä, että alihankkija ei voinut osallistua oman ryhmänsä tapahtumiin, lukuun ottamatta viikoittaisia tilannepäivytyspalavereita sekä projektiin suoraan liittyviin palavereihin. Näin ollen oman ryhmän tuttuus ei juuri kehittynyt vajaan parin vuoden aikana, joka ei ainakaan lisännyt luottamusta tai halukkuutta jakaa tai vastaanottaa tietoa.

Tapauksessa yksi organisaation kulttuuri ei juuri erotellut henkilöitä työnantajan mukaan vaan kaikilla oli lähes samankaltaiset oikeudet ja mahdollisuudet päästä käsiksi palveluihin ja dokumentaatioon. Omasta taustasta johtuen myös organisaation tavat toimia olivat tutumpia, joka helpotti myös viestintään liittyviä asioita.

Kieli yksittäisenä asiana oli molemmissa tapauksissa hieman haasteellista. Ensimmäisen tapauksen suurin ongelma liittyi aluksi toisten puheen ymmärtämisen vaikeuteen. Suullista viestintää vaikeutti varmasti myös yhteisen sanaston ja terminologian puuttuminen. Tapauksessa kaksi kielelliset ongelmat voidaan linkittää ainakin käytettyyn viestintämuotoon eli käytössä olleisiin sähköisiin viestimiin.

Työympäristö ja työkalut eivät eronneet toisistaan merkittävästi. Molemmissa tapauksissa käytössä oli avokonttorit ja viestintään käytetyt työkalutkaan eivät merkittävästi poikenneet toisistaan. Tapauksessa yksi käytettiin kuitenkin merkittävästi enemmän pikaviestimiä, kun taas tapauksessa kaksi puhelinta käytettiin enemmän. Puhelimen käytöllä pyrittiin mahdollisesti vähentämään viivettä viestinnässä. Ohjelmistokehityksessä käytetyt ohjelmat sen sijaan erosivat toisistaan, kun toisessa oli yleisesti käytössä olevat ohjelmistot ja toisessa ohjelmistossa ja käytännöissä näkyi hieman historian painolasti. Viestinnän suhteen ei voida työympäristöä eikä työkaluja kummassakaan tapauksessa nostaa toista suuremmaksi ongelmaksi.

Ryhmän tehokkuuden mittarina voidaan kai pitää lopputulosta. Ensimmäisen tapauksen osalta odotettu lopputulos oli saumaton töiden siirto ryhmien välillä, kun taas toisessa tapauksessa tavoiteltu lopputulos oli se, että useat ryhmät omaksuisivat yhteiset testauskäytännöt. Toisena ryhmän tehokkuutta kuvaavana asiana voidaan pitää sitä kuinka hyvin aikataulu piti. Kolmantena asiana ryhmän tehokkuutta tiedon siirtämisessä voidaan havainnoida sillä perusteella miten laadukkaasti asiat tehdään tiedon siirtämisen jälkeen. Molemmissa tapauksissa lopputulos saavutettiin, mutta ensimmäisessä tapauksessa pysyttiin paremmin aikataulussa, kun taas toisessa tapauksessa onnistuttiin laadullisesti paremmin.

Virtuaalitiimin käyttöä toisessa tapauksessa tehosti virtuaalitiimin vetäjän vierailut tiimin jäsenten luona. Vaikka ensimmäisessä tapauksessa ei virtuaalitiimiä ollutkaan, olisi siitä voinut olla jatkovaiheessa hyötyä, kun kolmen viikon visiitti oli ohi.

#### 4. Viestinnän tehostaminen hajautetussa ympäristössä

Tässä luvussa pohditaan, miten hajautettua työntekoa voidaan tehostaa tehostamalla ryhmän sisäistä ja ulkoista viestintää. Lisäksi pohditaan yksittäisiä asioita, joilla on merkitystä viestinnän onnistumiseen tai epäonnistumiseen ja miten näihin ongelmiin voidaan vaikuttaa. Eräs merkittävä ja näkyvä asia, johon huonolla viestinnällä on vaikutusta, on tuotteen lopullinen laatu ja toiminnollisuus. Asiakas näkee todennäköisesti ainoastaan tuotteen lopullisen version ja näin ollen siinä olevat ongelmat ja viat ovat osoitus puutteellisesta testaamisesta. Myöskin lopulliseen tuotteeseen päässeet viat ovat aina haasteellisempia ja kalliimpia korjata. Tässä luvussa pohditaan myös hieman sitä, miten tuotteessa olevat viat voitaisiin havaita mahdollisimman aikaisessa vaiheessa tehostamalla viestintää ja näin ollen säästää sekä kustannuksia, että ylläpitää yrityksen mainetta asiakkaisen silmissä.

Oletuksena on, että tässä kappaleessa pohditaan hajautettua ohjelmistotyötä ja viestintää yleisesti. Ohjelmistosuunnittelu on mitä suurimmassa määrin yhteistyötä vaativaa työtä jo luonnostaan. Laajan ohjelmiston ja järjestelmän toteuttaminen sekä testaaminen vaativat erittäin laajaa järjestelmätuntemusta, joka taas tarkoittaa tarvetta viestinnälle. Näin ollen on luontevaa, että testauksessa, ainakin välillisesti voidaan hyödyntää eri ohjelmistovaiheiden ammattilaisia, kuten tuoteasiantuntijoita, järjestelmäarkkitehteja, ohjelmistosuunnittelijoita, projektipäälliköitä jne. Lisäksi tässä luvussa pohditaan mahdollisuutta yhdistää eri menetelmistä parhaita puolia, jolloin saadaan tehokkaammin siirrettyä tietoa, tietämystä sekä osaamista. Loppujen lopuksihan viestinnän ensisijainen tarkoitus on välittää henkilön ajatukset ja ideat kohdeyleisölle mahdollisimman tehokkaasti niin, että tieto/tietämys välittyy ja vastaanotetaan tarkoituksenmukaisesti.

Seuraavassa muutamia yksittäisiä asioita, jotka vaikuttavat negatiivisesti viestintään sekä mahdollisia ratkaisuja ongelmiin.

Yleinen asenneilmapiiri hajautetun ohjelmistotuotannon eri toimipisteiden välillä ja miten sitä voitaisiin parantaa. Jos yleinen asenneilmapiiri eräässä toimipisteessä on kielteinen toisessa toimipisteissä työskenteleviä kohtaan, niin mistä se voi johtua? Yleisempiä selityksiä kielteisille asenteille ovat fyysinen etäisyys, kieli, kulttuurien väliset erot, yhteisen sosiaalisen maaperän puuttuminen, tietämättömyys, läpinäkyvyyden puute toisten tekemiseen jne. Voisiko esimerkin voimalla helpottaa muutosvastarintaa?

Oikean henkilöstön valinta hajautetussa ohjelmistosuunnittelussa on myös erittäin tärkeää. Miten oikea organisaatorakenne vaikuttaa hajautetussa

ohjelmistosuunnittelussa? Ainakin organisaatiossa tulee ottaa huomioon mahdolliset hierarkiasta johtuvat ongelmat.

Ammatillisuus ja vaikeneminen: ammattitaidon mittarina ainakin länsimaissa pidetään sitä, että henkilö kykenee tekemään itsenäisesti määrätyt tehtävät ilman, että muiden tarvitsee puuttua henkilön toimiin. Avun pyytämistä pidetään kenties heikkouden merkinä tai merkinä siitä, että henkilö ei ole ymmärtänyt mitä hänen pitäisi tehdä. Kenties henkilö on ymmärtänyt, mitä pitäisi tehdä, mutta ei syystä tai toisesta kykene suoriutumaan tehtävästä. Tämä saattaa johtaa siihen, että ongelmia ei myönnetä ja tämän johdosta päädytään tekemään huonoja ratkaisuita. Tokihan jokainen meistä haluaa olla ammatillisesti omavarainen? Ammattimaisuuden korostaminen ja ihannointi saattaa johtaa myös siihen, että kukaan ei uskalla myöntää omaa vajavaisuuttaan edes uusien ja ennestään tuntemattomien asioiden kohdalla. Uusien asioiden opetteluun menee jokaiselta yksilöllinen aika, riippuen siitä mitä, miten ja kuka asiaa on opiskelemassa. Jos jokainen joutuu opiskelemaan uudet asiat aina omin päin eikä asiasta käydä keskustelua edes oman ryhmän sisällä, se voi johtaa tilanteeseen jossa jokainen tulkitsee uudet asiat omalla tavallaan ja näin ollen samasta asiasta löytyy yhtä monta tulkintaa kuin ryhmässä on jäseniä. Tämä korostuu erityisesti hajautetussa ohjelmistosuunnittelussa, jossa spontaaneita keskusteluita ei esiinny luonnostaan. Näin ollen asia tulee helpommin haudatuksi ja oppiminen jää jokaisen oman harkinnan varaan. Mitä ammatillista osaamista korostaviin kulttuureihin tulee, lupa epäonnistua kuulostaa melkein kirosanalta. Mutta voisiko epäonnistumisesta oppia jotain? Esimerkiksi sen, että samojen virheiden toistaminen vuodesta toiseen ei ole kannattavaa? Tältä voidaan mahdollisesti välttyä, jos yleistä ilmapiiriä saadaan muokattua epäonnistumisia sallivampaan suuntaan sekä tarjoamalla mahdollisuuden oppia uutta esimerkiksi laajentamalla osaamista oman työn ohella tai tarjota omaa osaamistaan muille.

Yhtenä avaintekijänä viestinnän lisäämiseksi ja parantamiseksi tarjotaan virtuaalitiimin käyttämistä. Virtuaalitiimin tulisi koostua eri ohjelmistotuotannon osa-alueen asiantuntijoista, kuten esimerkiksi testauksen, toteutuksen ja määrittelyn osa-alueista. Mitä laajemmasta järjestelmästä on kyse, sitä tärkeämpää on, että jokaisesta järjestelmän osa-alueesta löydetään henkilö, jolla on kyseisestä osa-alueesta ja sovellusalueesta riittävä asiantuntemus sekä näkemys eri toiminnollisuuksista ja riippuvuuksista järjestelmän muihin osiin. Myös eri teknologioiden (kielet, työkalut jne.) osaamista tulee löytyä. Näiden tarkoitus on mahdollistaa ja yhtenäistää yrityksen sisällä käytössä olevia käytäntöjä kaikilla ohjelmistotuotannon osa-alueilla. Viestinnän kannalta hajautetussa ohjelmistoprojektissa tärkein yksittäinen asia on yksilö, joka aktiivisesti osallistuu tiedon ja tietämyksen jakamiseen, vierailee eri toimipisteissä, mahdollisesti pidempiäkin aikoja. Kasvokkain käytävät keskustelut hajautetun ryhmän kesken ovat elintärkeitä tietämyksen

ja tiedon välityksen sekä luottamuksen synnyttämisen kannalta. Kuitenkaan harvoin on mahdollista järjestää koko ryhmän vierailua toisessa toimipisteessä, esimerkiksi taloudellisista syistä. Tällöin henkilö, kutsuttakoon häntä ”sillanrakentajaksi” tai ”portinvartijaksi”, luo suhdeverkoston eri toimipisteiden kesken matkustamalla niihin sekä työskentelemällä niissä. Tämä helpottaa jatkossa viestintää kahden toimipisteen välillä, koska sillanrakentajan ja ryhmän välille on syntynyt luottamussuhde. Tietämyksen ja tiedon jakamisen kannalta olisi kuitenkin tärkeätä, että mahdollisimman usealla henkilöllä olisi mahdollisuus vierailla toisissa toimipisteissä ja viettää siellä aikaa. Esimerkkinä voisi ottaa vaikka tilanteen, jossa kahden ryhmän testaajat vaihtavat toimipisteitä ja käyvät jakamassa omaa osaamistaan sekä oppimassa toisilta hyviä käytäntöjä ja lisäämään järjestelmätuntemusta. Jatkossa esimerkiksi virtuaalisten ryhmien viestintä helpottuu, koska henkilöillä on vahvempi yhteinen tausta ja enemmän yhteistä maaperää jaettujen kokemusten myötä. Vierailut vähentävät myös epäluuloja esimerkiksi vieraisiin kulttuureihin sekä kieleen liittyviin uskomuksiin. Epäluulojen hälvetessä myös turhautuminen toisten työtä ja osaamista kohtaan lievenee. Näiden asioiden ansiosta työskentely yhteisten päämäärien eteen helpottuu, työn tehokkuus lisääntyy esimerkiksi siitä syystä, että henkilöt osaavat ennakoita, mitä toiset kykenevät tekemään ja mitä itseltä vaaditaan tavoitteiden toteuttamiseksi.

Nykyaikana erittäin suositun ilmiön, sosiaalisen median, valjastaminen yrityskäyttöön saattaisi tarjota yhden uuden vaihtoehdon yhteisöllisyyden lisäämiseksi hajautetuissa ohjelmistosuunnittelu projekteissa. Tähän liittyvät mm. käsitteet crowdsourcing sekä group sourcing. Näiden tarjoamat lisäarvot viestinnän osalta ovat aikakin vertaisryhmien välinen tietämyksen jakaminen ja yhteisöllisyyden luominen. Näistä ainakin group sourcingia voisi soveltuvien osin käyttää esimerkiksi testausfoorumina, jonka tarkoitus on jakaa järjestelmätietämystä mahdollisimman laajalle käyttäjäkunnalle oman organisaation sisällä, mutta myös soveltuvien osin sidosryhmille (asiakas, alihankkijat jne.). Testausfoorumin yhtenä tarkoituksena on löytää järjestelmän eri osa-alueissa piileviä ongelmia mahdollisimman aikaisessa vaiheessa, mahdollisimman hyvissä ajoin ennen kuin tuote päättyy asiakkaalle. Mahdollisena motivaatioina osallistua testausfoorumin aktiviteetteihin voisivat olla mahdollisuus löytää vastauksia omiin haasteellisiin ongelmiin sekä mahdollisuus tarjota omaa ammattiosaamista muiden käyttöön. Se tarjoaa myös mahdollisuuden kartuttaa omaa järjestelmäosaamista. Ongelmia testausfoorumin käyttöönotolle löytyy varmasti paljon. Ensinnäkin uuden työkalun käyttöönoton synnyttää aina vastarintaa ja näin ollen käyttöönottovaiheeseen tulee panostaa aikaa, resursseja ja rahaa. Käyttöönottoa tulee aktiivisesti seurata ja mukauttaa käyttäjien ehdoilla, näin sitoutetaan käyttäjät työkalun käyttäjiksi. Käyttöönoton jälkeen on edelleen jatkettava työkalun kehittämistä sekä pyrittävä löytämään parempia käytäntöjä työkalussa ilmenneiden ongelmien pohjalta. Vapaaehtoisuudessa tämän tyyppisessä

työkalussa on sekä hyvät että huonot puolensa. Hyvänä puolena on se, että käyttäjät, jotka vapaaehtoisesti käyttävät foorumia ovat todennäköisesti motivoituneita sekä hakemaan apua, että tarjoamaan sitä muille. Huonona puolena voidaan olettaa, että esimerkiksi testausfoorumi ei varmastikaan saavuta kaikkia mahdollisia sidosryhmiä vapaaehtoisuutensa vuoksi. Tämän vuoksi uusi -työkalu tulisi integroida muuhun sovellusympäristöön, että sen käyttö olisi mahdollisimman vaivatonta.

Yhtenä tärkeänä työkaluna on nostettava esille eri testausvaiheiden automatisointi ja niistä saatujen testaustulosten jakaminen eri sidosryhmille. Tämä edustaa viestinnän osalta tiedonjakamista. Tiedonjakamisen osalta tärkeitä ovat ainakin seuraavat asiat: tuloksia jaettaessa laajemmalle kohdeyleisölle, tulee tulosten olla yksiselitteisiä, riittävän kuvaavia sekä tarkoituksen mukaista. Jos testitulosten seuraaja on esimies, eivät yksityiskohtaiset kuvaukset eivät ole tarpeellisia. Jos taas seuraaja on testaaja, tärkeämpää on saada ymmärrys, mistä ominaisuudesta on kysymys. Kun taas testituloksia tutkii ohjelmoija, niin tärkeintä on pystyä tulosten perusteella kohdentamaan ongelma mahdollisimman helposti ja tarkasti koodissa oikeaa paikkaan. Jokaisesta testausvaiheesta (yksikkö, moduuli, integrointi, järjestelmä) havaitut virheet tulisi analysoida sillä tarkkuudella, että löydetään virheeseen johtanut syy. Tiedot tallennetaan esimerkiksi bug tracking -työkaluun tms. jatkoa varten. Analysoitujen tietojen pohjalta on mahdollista löytää puutteita prosesseista ja parantaa niitä. Mitä pidemmälle tuotannon elinkaareissa viat pääsevät, sitä kalliimmiksi ne tulevat. Pelkkä tiedon analysointi ei vielä riitä, vaan sen pohjalta pitää saada aikaan muutoksia prosesseihin ja/tai käytäntöihin. Jotta muutokset prosesseissa ja/tai hyväksi havaituissa käytännöissä saada kaikkien tietoon, on muutoksista tiedotettava sidosryhmille ja järjestettävä tarvittaessa koulutusta.

Voidaanko avoimen lähdekoodin (OSS) tavasta kehittää tuotteita, löytää hyviä käytäntöjä miten viestintä järjestetään, myös suljetun ohjelmiston (CSS?) puolelle?

Mahdollisuus järjestää yhteinen testausalustaa, jossa organisaation eri osastot pääsevät tutustumaan koko järjestelmän toiminnollisuuden testaamiseen? Esimerkiksi virtuaalinen testausympäristö, jota voidaan kopioida helposti?

Seuraavaksi käydään läpi, miten kokonaisuudessaan saataisiin paras hyöty erilaisista työmenetelmistä ja niiden yhdistelmistä, kun lähtökohtana on tuotteen laadun parantaminen. Tässä tuotteen laadun parantamisella tarkoitetaan sitä, että koko järjestelmän tasolla vikojen määrä voitaisiin vähentää minimiin järjestelmän elinkaaren siinä vaiheessa, jossa sitä jo toimitetaan asiakkaille. Toisin sanoen pyritään löytämään paras mahdollinen yhdistelmä eri työmenetelmistä, jolla viat kyetään löytämään mahdollisimman aikaisessa vaiheessa järjestelmän elinkaarta. Työmenetelmistä



tärkeimpinä ovat virtuaalitiimin käyttöönotto, tietämyksen, tiedon ja osaamisen tehokkaampi siirtäminen organisaation sisällä ja soveltuvien osien myös organisaatiosta ulospäin, testauksen automatisointi ja testitulosten jakaminen, testausympäristöjen tehokas käyttö sekä mahdollisuuksien mukaan pyrkiä hyödyntämään muitakin sidosryhmiä määrittelyvaiheessa sekä testaamisessa.

Ensimmäinen ja kenties tärkein tehtävä on määrittellä virtuaalitiimin tehtävä. Virtuaalitiimin tehtävien määrittely lähtee aina jostakin tarpeesta. Tässä tapauksessa on ajan saatossa vähentää vikojen kokonaismäärää järjestelmässä, mutta ensisijainen tarkoitus on havaita viat mahdollisimman aikaisessa vaiheessa tuotteen elinkaarta. Kuten jo useampaan kertaan aiemmin on todettu, niin mitä aiemmin viat havaitaan, sitä helpompi ja halvempi korjaus yleensä on. Mitä myöhemmin vika taas havaitaan, sitä haastavampaa ja kalliimpaa sitä on korjata. Vikojen havaitseminen vasta siinä vaiheessa, kun tuote on jo asiakkaalla saattaa vaikuttaa negatiivisesti myös yrityksen maineeseen. Virtuaalitiimejä voidaan perustaa useampiakin eri tarkoituksia varten, kuten eri ohjelmistosuunnitteluvaiheiden tarpeisiin. Seuraavassa yleinen lista mahdollisista virtuaalitiimin tehtävistä:

- Muodostaa verkosto järjestelmän eri osa-alueiden asiantuntijoista.
- Jakaa tietoa, tietäystä ja osaamista.
- Tarjota laaja asiantuntijuus ratkaista haasteellisiakin ongelmia.
- Yhtenäistää toteutusvaiheen työt.
- Yhtenäistää testausvaiheet sekä testausmenetelmät järjestelmän laajuisesti.
- Yhtenäistää kehitystyökalut sekä testaukseen työkalut koko järjestelmän laajuisesti.
- Tarjota testaustuloksia järjestelmän laajuisesti.
- Testausympäristöjen laajempi käyttö.

Verkostoituminen eri osa-alueiden asiantuntijoiden kanssa luo vankan perustan viestinnälle virtuaalitiimin jäsenten kesken. Erittäin tärkeätä olisi järjestää myös vähintään yksi tapaaminen (kasvokkain) heti, kun virtuaalitiimi perustetaan. Tällä pyritään madaltamaan etäisyyden, aikaeron tuomia haasteita viestintään. Jos tiimin jäsenten kesken pystytään rakentamaan parempi luottamussuhde, niin se on omiaan ehkäisemään esimerkiksi ristiriitatilanteita ja ristiriitatilanteista selviytymistä. Vaikka yhteiset tavoitteet sitovat virtuaalitiimin jäseniä yhteen, niin myös yhteiset kasvokkain järjestettävät tapaamiset auttavat laajentamaan yhteistä kokemuspohjaa, joka omalta osaltaan auttaa kasvattamaan tiimin yhteishenkeä. Verkostoitumisen tueksi tulee myös tarjota riittävät työkalut, kuten sähköpostilistat, keskustelupalstat, pikaviestimet sekä videoneuvottelun mahdollistavan tehokkaan viestinnän.

Tiedon, tietämyksen ja osaamisen jakaminen virtuaalitiimin jäsenten kesken on erittäin tärkeä osa tuotteen laadun parantamisessa. Tiedon ja tietämyksen jakaminen virtuaalitiimissä vaatii tarkemmin määriteltäviä sääntöjä, jotta jokainen kykenee tuottamaan sellaista kirjallista dokumentaatiota, että muut tiimin jäsenet ymmärtävät asiat kuin ne on tarkoitettu. Yhteisen kielen määrittelemisen, kuten yhteinen termistö ja lyhenteet auttavat ehkäisemään väärinkäsityksiltä. Esimerkiksi transaktiivisen muistin käyttöönottoa kannattaa harkita. Tiedon ja tietämyksen jakaminen tehostuu, jos on mahdollista lähettää tiimin jäseniä tutustumaan toisten tiimien jäseniin ja toteuttamaan tai testaamaan eri järjestelmän osia. Tämä auttaa paremmin hahmottamaan järjestelmää kokonaisuutena.

Mitä laajempi järjestelmä, sitä haasteellisempia ovat yleensä myös ongelmat. Myös asiantuntijuus laajoissa järjestelmissä on usein hajautunut ja juuri näiden asiantuntijoiden kokoaminen yhteen mahdollistaa haasteellistenkin ongelmien ratkaisemisen. Haasteellisimpia ongelmia ovatkin usein ne, jotka ulottuvat koko järjestelmään. Jos asiaa ajatellaan vielä hieman laajemmin, niin mahdollisuuksien mukaan myös asiakkaan ottaminen mukaan saattaa tuoda uusia ajatuksia järjestelmän käyttöön liittyen ja sitä kautta järjestelmässä mahdollisesti piileviin ongelmiin.

Testausvaiheiden yhtenäistämällä tarkoitetaan muun muassa sitä, mitä missäkin vaiheessa testataan.

Testaustyökalujen yhtenäistämällä voidaan saavuttaa useitakin etuja. Ensinnäkin tiimin jäsenien vierailusta saadaan enemmän irti, jos osaa ajasta ei tarvitse käyttää työkalujen käytön opiskeluun. Samoin parhaiden käytäntöjen jakaminen organisaation sisällä helpottuu, kun käytössä on samat työkalut. Järjestelmätestauksessa on helpompaa ottaa käyttöön muiden tekemiä testejä.

Testaustulosten jakaminen tarjoaa hyvän näkymän järjestelmän testaamisen tilaan. Tulokset tulisi esittää siten, että niistä voivat hyötyä niin johtotehtävissä olevat kuin ruohonjuuritasolla työskentelevät henkilöt. Ongelmia ratkottaessa on helpompaa näyttää miten kyseinen ominaisuus on testattu järjestelmässä. Testaustulosten perusteella voidaan mahdollisesti löytää suoraan kohta, jossa kyseinen ominaisuus on mennyt rikki tai ainakin se tarjoaa hyvän lähtökohdan jatkotutkimuksille.

Testausympäristöjen käyttö aiheuttaa yleensä ongelmia, koska niitä on tarjolla liian vähän. Tämä saattaa johtua esimerkiksi siitä, että ohjelmistosta pitää olla useita eri versioita käytössä, jolloin tietyn vian havaitseminen saattaa vaatia jotain tiettyä ohjelmistoversiota. Ohjelmistoversioiden hallintaan sekä tehokkaampaan testiympäristöjen käyttöä varten on

syytä harkita automatisoituja käännöskoneita sekä automatisoitavia testejä, jolloin on helppo määritellä millä ohjelmistoversiolla ja mitä halutaan testata. Kun testit on ajettu, saa käyttäjä tiedon sekä tulokset niiden lopullista manuaalista analysointia varten.

Asiakasviat ovat hyvä esimerkki niistä vioista, joita on syytä analysoida huolellisemmin. Asiakasvikojen analysointiin voidaan käyttää esimerkiksi Root Cause Analysis (RCA) sekä Escape Defect Analysis (EDA) analyyssejä, joilla pyritään kartoittamaan missä ja minkä vaiheen testeissä kyseinen vika olisi pitänyt havaita. Matalimman tason testeissä katselmoinnit sekä parikoodaus ovat hyviä työkaluja parantamaan testien kattavuutta. Myös erilliset testausmallit kuten TDD tai MBT ovat myös jossain tapauksissa hyviä apuvälineitä. TDD auttaa miettimään koodin rakennetta tarkemmin, jolloin koodiin ei välttämättä tule turhia haaroja. MBT taas omalta osaltaan auttaa lisäämään testien kattavuutta.

Testauksen automatisointi on edellytys sille, että testituloksia on mahdollista ajaa mahdollisimman helposti ja toistaa tarvittaessa vaikka kuinka usein.

Testitulosten jakaminen on omalta osaltaan hyvä tapa lisätä tiedon jakamista virtuaalitiimin kesken.

Testausympäristöjen käyttö niin, että koko projekti hyötyy siitä. Laajassa järjestelmässä usein hämärtyy miten rajapintoja tarjoavien palveluiden kuuluisi toimia ja miten niitä testataan kattavasti. Testauksessa on usein käytössä simulaattoreita tai muita korvaavia toteutuksia. Jos toteutus muuttuu, niin päivitetäänkö testauksessa käytetyt simulaattorit ja muut korvaavat toteutukset? Jos tarjolla olisi referenssitoteutus rajapinnin toteutuksesta ja sitä vasten olisi mahdollista testata ainakin osa testeistä. Virhetilanteiden kannalta simulaattorit jne. ovat kuitenkin tarpeellisia.

## 5. Päätelmät ja yhteenveto

Tässä luvussa vedetään yhteen tutkielman tärkeimmät huomiot sekä vastataan tutkimuskysymykseen ” *Mitä ovat hajautetun ohjelmistosuunnittelun ongelmat viestinnässä ja miten niitä voidaan ehkäistä?* ”

Tärkein huomio oli, että mitään yksittäistä asiaa ei voida nostaa selkeästi muita tärkeämmäksi, kun yritetään ehkäistä hajautetussa ohjelmistosuunnittelussa havaittuja ongelmia. Viestintä on erittäin laaja aihe ja sen laatuun vaikuttavat monet asiat, kuten tapaustutkimuksissakin on havaittu. Viestintä on siis yksi tärkeimmistä asioista, joka aiheuttaa paljon ongelmia ja joka toimiessaan ratkaisee useita ongelmia. Toisaalta koska viestintä on erittäin laaja aihealue, johon kiinteästi liittyvät kulttuuri ja kieli, mutta myös etäisyys sekä viestintään käytetty tekniikka. Kulttuurieroihin suhtaudutaan usein kovin tunnepitoisesti. Vaikka kulttuurieroista johtuvia ongelmia ei pidä mitenkään väheksyä, voitaisiin ongelmista usein selvitä käyttämällä maalaisjärkeä. Puhumattakaan, jos projektiin saataisiin ripaus inhimillisyyttä ja empatiaa. Tuskinpa niilläkään yksistään mitään ongelmia saataisiin ratkottua, mutta sillä voitaisiin luoda hedelmällisempi maaperä esimerkiksi luottamuksen ja yhteishengen kasvulle. Organisaation näkökulmasta erittäin tärkeä pääoma on henkilöstön osaaminen ja toisaalta ongelmana se, miten organisaatio onnistuu valjastamaan kaiken sen osaamisen koko organisaation hyväksi. Onnistuessaan tietämyksenhallinta mahdollistaa esimerkiksi hiljaisen tiedon käyttöön ottamisen mahdollisimman tehokkaasti. Eräs tärkeä yksittäinen asia on se, miten ryhmään saadaan jokaiseen tehtävään soveltuva henkilö, tämä pätee erityisesti kun kyseessä on hajautettu ohjelmistoprojekti. Kun henkilöitä valitaan hajautettuun ohjelmistoprojektiin, tulisi valinnassa kiinnittää mahdollisuuksien mukaan myös henkilön sosiaalinen kyvykkyyteen, ei vain pelkkään tekniseen osaamiseen. Tosi elämässä näitä ”hyviä tyyppejä”, joiden tekninen osaaminen on riittävällä tasolla, ei ole ruuhkaksi asti.

Voidaanko hajautetun ohjelmistoprojektin ongelmia lieventää esimerkiksi jakamalla organisaation sisällä käytössä olevia hyviksi havaittuja käytäntöjä myös organisaation sidosryhmiin ja päinvastoin?

Ei ole mitään syytä epäillä ettei näin olisi. Se miten yhteistyössä hajautetussa ympäristössä onnistutaan, riippuu täysin kyseessä olevan organisaation käytännön toimista tai vielä tarkemmin miten yksittäisistä projekteista voidaan oppia. On varmaa, että osa projekteista onnistuu ja osa epäonnistuu. Tärkeämpää kuin yksittäiset onnistumiset tai epäonnistumiset olisi oppia omista virheistään, sillä virheet ovat parhaimmillaan kuin

erinomaisia oppitunteja. Jos virheitä halutaan välttää kokonaan, on parempi olla tekemättä mitään.

Olipa organisaation ongelma sitten mikä tahansa, vaikkapa hajautetun ohjelmistosuunnittelun haasteet, niin jokaisen organisaation on tehtävä omat ratkaisunsa. Mitään valmista yleispätevää ratkaisua ei ole. Pyörää ei kuitenkaan kannata keksiä uudelleen, organisaatio itsessään sisältää usein valtavan määrän valmiita ratkaisuita tai samankaltaisiin ongelmiin voidaan löytää ratkaisuja myös organisaation ulkopuolelta, kunhan ne vain osataan soveltaa oman organisaation tarpeisiin sopiviksi. Toisaalta toimivaa ei kannata rikkoa eli muutosta ei kannata tehdä muutoksen vuoksi, vaan muutoksen täytyy lähteä aina tarpeesta. Harvemmin muutokset ovat myöskään pysyviä ja kestävät muuttumattomina pitkään, vaan muutosten vaikutuksia ja toimivuutta täytyy seurata jatkuvasti ja reagoida niihin tarpeen mukaan.

## Viiteluettelo

[Bacchelli and Bird, 2013] Bacchelli, Alberto and Bird, Christian. Expectations, outcomes, and challenges of modern code review. In: Proceedings of the 2013 International Conference on Software Engineering. IEEE Press, 2013. p. 712-721.

[Beck, 2003] Beck, Kent. Test-driven development: by example. Addison-Wesley Professional, 2003.

[Bikram *et al.*, 2006] Sengupta, Bikram, Satish Chandra and Vibha Sinha. A research agenda for distributed software development. In: Proceedings of the 28th international conference on Software engineering. ACM, 2006. p. 731-740

[Boden and Avram, 2009] Boden, Alexander; Avram, Gabriela. Bridging knowledge distribution-The role of knowledge brokers in distributed software development teams. In: ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 2009. CHASE'09.. IEEE, 2009. p. 8-11.

[Boden *et al.*, 2009] Boden, Alexander, Avram, Gabriel, Bannon, Liam and Wulf Volker. Knowledge Management in Distributed Software Development Teams-Does Culture Matter? In: proceeding of the Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on. IEEE, 2009. p. 18-27

[Brook, 1975] Frederick P. Brook, Jr. The Mythical Man-Month, Essays of Software Engineering. Addison-Wesley Publishing Company, Inc. 1975.

[Cataldo and Herbsleb, 2008] Cataldo, Marcelo, and James D. Herbsleb. Communication networks in geographically distributed software development. In: Proceedings of the 2008 ACM conference on Computer supported cooperative work. ACM, 2008. p. 579-588

[Chamberlain, 2014] Chamberlain, Jon. Groupsourcing: Problem Solving, Social Learning and Knowledge Discovery on Social Networks. In: Proceeding of the Second AAAI Conference on Human Computation and Crowdsourcing. 2014.

[Cockburn and Williams, 2000] Cockburn, Alistair and Williams, Laurie. The costs and benefits of pair programming. Extreme programming examined, 2000. p. 223-247.

[Erran and Tjia 2005] Carmel, Erran and Paul Tjia. Offshoring Information Technology, Sourcing an Outsourcing to a Global Workforce. Cambridge University Press, 2005.

[Espinosa *et al.*, 2007] Espinosa, J. Alberto, Slaughter, Sandra A. and Kraut, Robert E. and Herbsleb, James D. Familiarity, complexity, and team performance in geographically distributed software development. *Organization science* 2007, 18.4: 613-630.

[Espinosa *et al.*, 2007] Espinosa, J. Alberto, Slaughter, Sandra A. and Kraut, Robert E. and Herbsleb, James D. Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems* 2007, 24.1: 135-169.

[Estellés-Arolas and González-Ladrón-De-Guevara, 2012] Estellés-Arolas, Enrique; González-Ladrón-De-Guevara, Fernando. Towards an integrated crowdsourcing definition. *Journal of Information science*, 2012, 38.2: 189-200.

[Gerrit, 2015] Gerrit, web based code review and repository management for the Git version control system, <https://www.gerritcodereview.com/>.

[Guinan, Coopridge and Faraj, 1998] Guinan, Patricia J., Coopridge, Jay G. and Faraj, Samer. Enabling software development team performance during requirements definition: A behavioral versus technical approach. *Information Systems Research*, 1998, 9.2: 101-125.

[Gutwin *et al.* 2004] Gutwin, Carl, Reagan Penner, and Kevin Schneider. Group awareness in distributed software development. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. ACM, 2004. p. 73-81.

[Hakkarainen, 2003], Hakkarainen, Kai. Kollektiivinen älykkyys. *Psykologia*, 2003, 38.6: 384-401.

[Haruhiko and Motoshi, 2006] Kaiya, Haruhiko; Saeki, Motoshi. Using domain ontology as domain knowledge for requirements elicitation. In: *Proceeding of the 14th IEEE International Conference. Requirements Engineering*, IEEE, 2006. p. 189-198.

[Hinds and Weisband, 2003] Hinds, Pamela J.; Weisband, Suzanne P. Knowledge sharing and shared understanding in virtual teams. *Virtual teams that work: Creating conditions for virtual team effectiveness*, 2003, 21-36.

[IRC, 2015] Internet Relay Chat, <http://fi.wikipedia.org/wiki/IRC>, 2015.

[Jenkins, 2008] Nick Jenkins. A Software testing Primer, An Introduction to Software Testing. Nick Jenkins, 2008

[Jiménez *et al.*, 2009] Jiménez, Miguel, Mario Piattini, and Aurora Vizcaino. Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering* 2009: 3.

[Kaurismäki, 2011] Kaurismäki, Aki. <http://yle.fi/aihe/artikkeli/2011/05/19/aki-kaurismaki-sanojen-tuolla-puolen>, 2011.

[Kim and de Dear 2013] Jungsoo Kim and Richard de Dear; Workspace satisfaction; The privacy-communication trade-off in open-plan offices. *Journal of Environmental Psychology* 2013, 36: 18-26.

[Komi-Sirviö ja Tihinen, 2005] Komi-Sirviö, Seija, and Maarit Tihinen. Lessons learned by participants of distributed software development. *Knowledge and Process Management* 2005, 12.2: 108-122.

[Lee *et al.*, 2000]. Lee, Jae-Nam, Huynh, Minh Q., Ron Chi-wai, Kwok and Pi, Shih-Ming. The evolution of outsourcing research: what is the next issue? *System Sciences*, In: *Proceedings of the 33rd Annual Hawaii International Conference on. IEEE*, 2000. p. 1-10.

[Lehto, 2012] Lehto, Jouni. kaksi näkökulmaa organisaatiotutkimuksen kulttuurisotaan. Tampereen yliopisto, 2012

[Lipnack and Stamps, 2000] Lipnack, Jessica and Stamps, Jeffrey. *Virtual Teams, People Working Across Boundaries with Technology*, second edition. John Wiley & Sons, inc., 2000.

[Martikainen, 2014] Martikainen, Jarkko, 2015. *Virheet on tehtävä itse. Koirien taivas*, Sakara Rekords, 2014.

[Narayan and Balan, 2007] Ramasubbu, Narayan, and Rajesh Krishna Balan. Globally distributed software development project performance: an empirical analysis. In: *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. ACM, 2007. p.125-134.



[Oshri *et al.*, 2009]. Ilan Oshri, Julia Kotlarsky, Professor Leslie P. Willcocks. The Handbook of Global Outsourcing and Offshoring. Palgrave Macmillan, 2009.

[Oshri *et al.*, 2008] Ilan Oshri, Paul Van Fenema and Julia Kotlarsky, "Information Systems Journal", 2008, 18, 593–616

[Patton, 2001] Patton, Ron, Software testing. Sams Publishing 2001.

[Paasivaara, 2005] Paasivaara, Maria. Communication practices in inter-organisational product development. Helsinki University of Technology, 2005.

[Ryan *et al.*, 2009] Ryan, Sharon and O'Connor, Rory V. Development of a team measure for tacit knowledge in software development teams. Journal of Systems and Software, 2009, 82.2: 229-240.

[Slaughter, 1998] Sandra Slaughter, Evaluating the cost of software quality, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States Communications of the ACM (Impact Factor: 3.62). 1998, 41(8):67-73.

[Vandermark, 2003] Mary Ann Vandermark. Defect Escape Analysis: Test Process Improvement. IBM Software Group, 2003.

[Wegner, 1987] Wenger, Daniel M. Transactive memory: A contemporary analysis of the group mind. In: Theories of group behavior. Springer New York, 1987. p. 185-208.